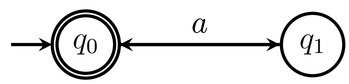
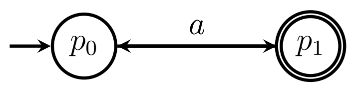
Формальные языки

домашнее задание до 23:59 05.03

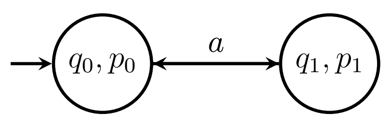
1. Доказать или опровергнуть утверждение: произведение двух минимальных автоматов всегда дает минимальный автомат (рассмотреть случаи для пересечения, объединения и разности языков).

   Это не так. Для объединения и пересечения рассмотрим автоматы для языков a^{2n} и a^{2n+1}:

   После удаления недостижимых вершин произведение будет выглядеть следующим образом (без расставленных терминальных состояний):

   Но минимальный автомат будет состоять из одной вершины: в случае объединения это должна быть петля, которая все принимает, а в случае пересечения это должна быть петля, которая все отвергает.

   Для разности надо взять оба автомата равными a^{2n}. Тогда их произведение будет выглядеть так же, а минимальный автомат будет петлей, которая все отвергает.

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

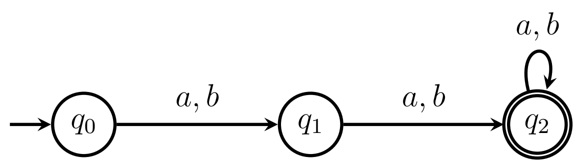
   (c) Минимальный полный детерминированный конечный автомат

То, что находится в центре, можно вообще выкинуть, потому что оно может быть пустым, в таком случае принимаются все слова длины хотя бы два, а в принятии слов меньшей длины этот центр не поможет.

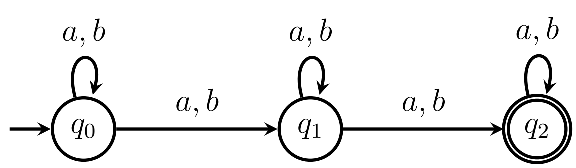
a. В пункте c.

b. В пункте c.

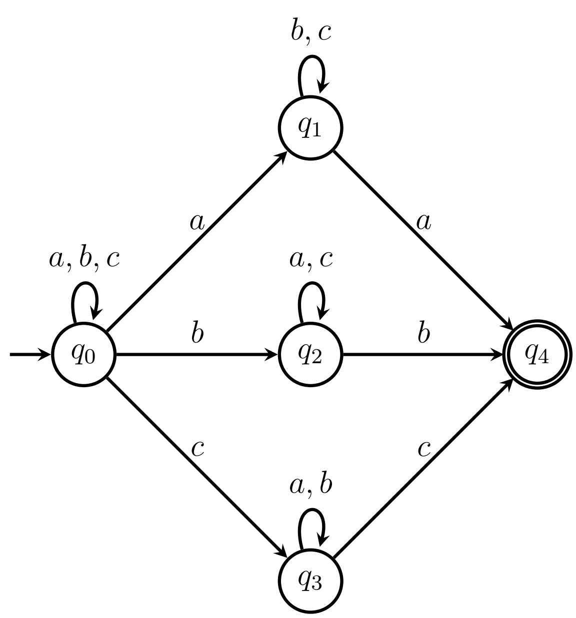
c. Полный автомат из одной вершины либо все принимает, либо все отвергает, так что он нам не подходит. Если же автомат состоит из двух вершин, посмотрим, какие вершины в нем терминальные. Если изначальная вершина терминальная, то примется пустое слово; если же терминальная вторая вершина, то в нее можно попасть по слову длины один, что нам тоже не подходит. Так что минимальный автомат имеет размер не меньше 3. Построим автомат из 3 вершин:



Можно еще недетерминированный… Я не очень понимаю задание.



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



Построим регулярное выражение для каждой вершины по отдельности начиная с начальной.

q_0 — $(a|b|c)^*$

q_1 — $(a|b|c)^*a(b|c)^*$

q_2 — $(a|b|c)^*b(a|c)^*$

q_3 — $(a|b|c)^*c(a|b)^*$

q_4 — $(a|b|c)^*((a(b|c)^*a)|(b(a|c)^*b)|(c(a|b)^*c))$

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

Пусть автомат существует. Пусть в нем n вершин.

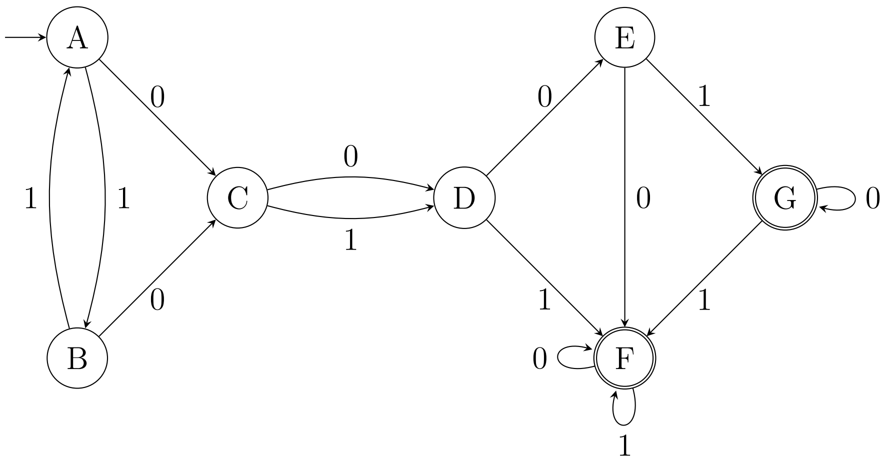
Рассмотрим строки вида 0^k1 для k от 0 до n. По принципу Дирихле две из этих строк закончат свое путешествие по автомату в одном и том же состоянии. Пусть это 0^a1 и 0^b1. После чего если мы еще пройдемся из этой вершины по строке 10^a, то должны попасть в терминальное состояние, потому что 0^a110^a — палиндром. Но, с другой стороны, строка 0^b110^a не является палиндромом, но при этом пройдет по тому же самому пути, так что придет в терминальное состояние. 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.

Сделаем аналогично предыдущей задаче. Будем рассматривать строки вида b^k. Какие-то две попадут в одно состояние. Пусть это b^x и b^y, при этом $x > y$. Тогда дополним строкой $aa(ba)^x$. Строка $b^xaa(ba)^x$ удовлетворяет условию, так что мы должны были попасть в терминальное состояние, но, с другой строоны, строка $b^yaa(ba)^x$ не подходит под условие (у нее только одно место, где две буквы a подряд, при этом слева y букв b, что меньше, чем x букв a справа), но при этом пройдет по тому же самому пути, так что придет в терминальное состояние. 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: