

Домашнее задание №3

Томинин Ярослав, 778 группа

23 сентября 2018 г.

1.

1)

$$Q = q_0, q_1, q_2$$

$$\Sigma = \{0, 1\}$$

$$\delta(q_0, 0) = q_1, \delta(q_0, 1) = q_1, \delta(q_1, 0) = q_2, \delta(q_1, 1) = q_0, \delta(q_2, 0) = q_1, \delta(q_2, 1) = q_2$$

$$q_0 = q_0$$

$$F = \{q_1\}$$

$$Q = q_0, q_1, q_2$$

$$\Sigma = \{0, 1\}$$

$$\delta(q_0, 0) = q_1, \delta(q_0, 1) = q_1, \delta(q_1, 0) = \{q_0, q_2\}, \delta(q_2, 0) = q_1, \delta(q_2, 1) = q_2$$

$$q_0 = q_0$$

$$F = \{q_1\}$$

2)

А-является, так как из каждого состояния есть две стрелки 0,1 и мы можем единственным образом перейти в другое состояние, нехависимо от входа

В-не является, так как из q_1 есть две стрелки с 0 и попав в него из q_0 мы не можем определенно перейти только в 1 состояние.

3)

Находимся в q_0 , принимаем 0, переходим в 0, принимаем 1, переходим в 1, принимаем 1 переходим в 0, принимаем два 0 переходим два раза в 0, принимаем 1, переходим в 1. Состояние допускающее, следовательно, слово допустимо.

4)

Находимся в 0, принимаем 0, переходим в 0, принимаем 1, переходим в 1, принимаем 0, переходим в 0,2, принимаем 1 переходим в 1,2, принимаем 0, переходим в 0,1,2, принимаем 0, переходим в 0,1,2, принимаем 1, переходим в 1,2. Есть допустимое состояние, поэтому принимает.

5)

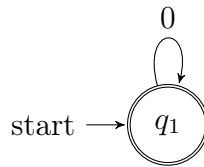
111 принадлежит А(из 0 в 1, потом в 0, потом в 1-принимающее состояние), 10 не принадлежит А(из 0 в 1, потом в 2-не принимающее)

0 не принадлежит В(из 0 в 0-не принимающее), 1-принимающее(из 0 в 1-принимающее)

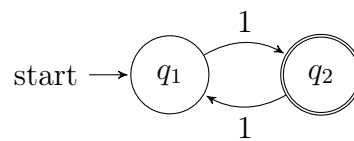
2.

Нет.

Автомат А:

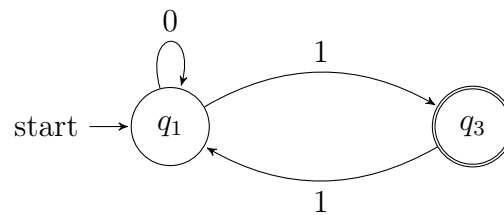


Автомат В:



А допускает слова из 0, а В, как мы уже знаем, из нечетного числа 1, их конкатенация должна состоять из слов с каким-то количеством 0, а потом нечетным количеством 1.

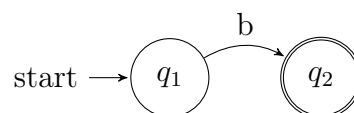
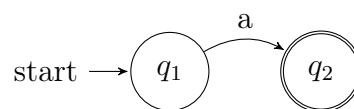
Докажем от противного: построим автомат и придем к противоречию. У нас получается



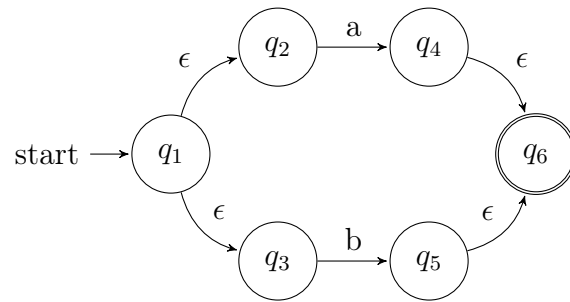
Оу, но тогда мы можем получить слово 11001, которое не принадлежит конкатенации. Противоречие. Доказанно.

3.

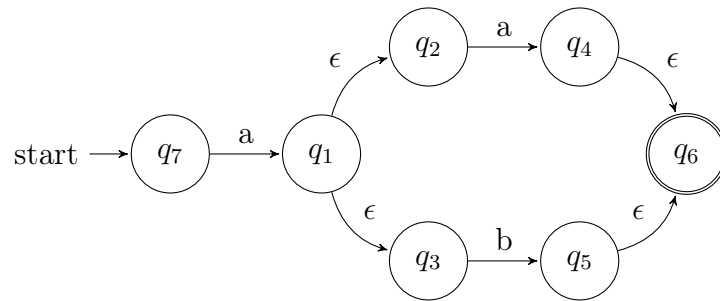
Имеем:



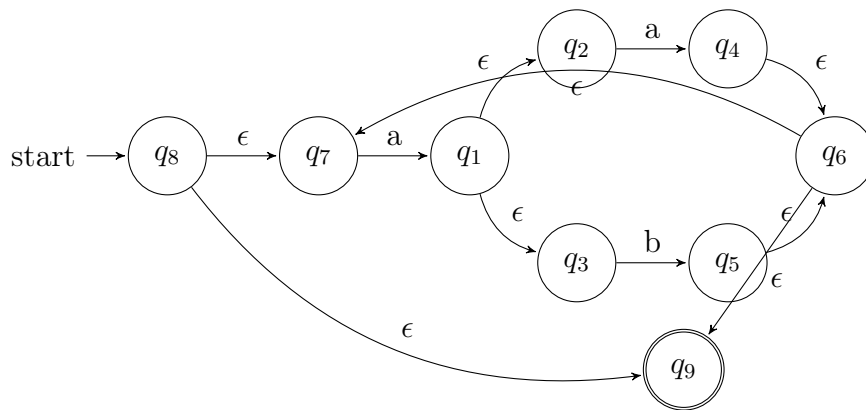
$a|b$



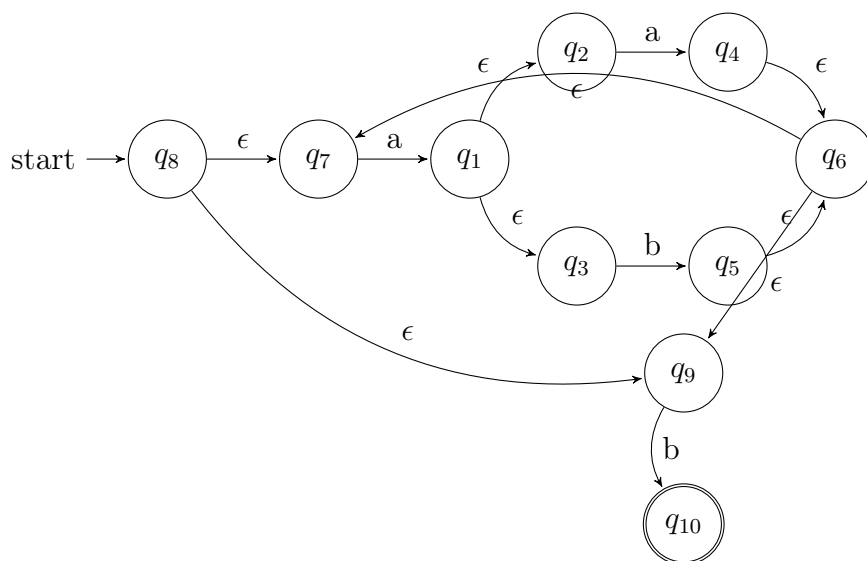
$a(a|b)$



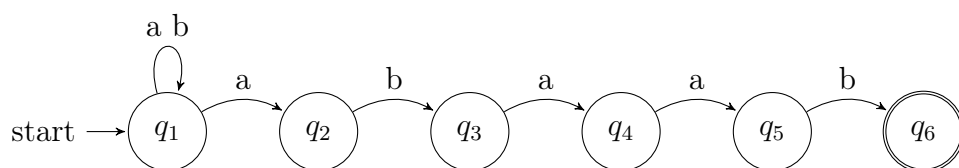
$(a(a|b))^*$



$(a(a|b))^*$



4.



Докажем, что этот автомат распознает только слова, оканчивающиеся на суффикс $abaab$. Для этого докажем инварианты:

- 1) когда мы попадаем в 2, то последовательность оканчивается на a (так как мы можем попасть в него только из 1 по a , то это очевидно)
- 2) когда мы попадаем в 3, то последовательность оканчивается на ab (так как мы можем попасть в него только из 2 по b и если используем 1 инвариант, то это очевидно)
- 3) когда мы попадаем в 4, то последовательность оканчивается на aba (так как мы можем попасть в него только из 3 по a и если используем 1 и 2 инварианты, то это очевидно)
- 4) когда мы попадаем в 5, то последовательность оканчивается на $abaa$ (так как мы можем попасть в него только из 4 по a и если используем 1, 2, 3 инварианты, то это очевидно)
- 5) когда мы попадаем в 6, то последовательность оканчивается на $abaab$ (так как мы можем попасть в него только из 5 по b и если используем 1, 2, 3, 4 инварианты, то это очевидно)

Следовательно мы доказали, что этот автомат распознает только слова, оканчивающиеся на суффикс $abaab$.

Докажем, что автомат распознает все слова, оканчивающиеся на abaab. От противного, допустим, что существует слова, оканчивающееся на abaab, но его автомат не распознает. Тогда мы рассмотрим слово w, как $w = u \cdot abaab$ и будем переходить в 1 состояние, пока будут подаваться символы из u. Мы так сможем сделать, потому что у нас есть переход из 1 в 1 по любому символу. Далее просто перейдем в 2,3,4,5,6 по символам abaab. Мы очутились в допустимом состоянии и пришли к противоречию. Следовательно, автомат распознает все слова, оканчивающиеся на abaab.

5.

Рассмотрим $Q_1 = \epsilon - \text{closure}(q_1) = q_1$

Q_1 переход по a(Q_2)

$Q_2 = \epsilon - \text{closure}(q_1) | \epsilon - \text{closure}(q_2) = \{q_1, q_2\}$

Q_1 переход по b(Q_1)

$Q_3 = \epsilon - \text{closure}(q_1) = q_1 = Q_1$

Q_2 переход по a(Q_2)

$Q_4 = \epsilon - \text{closure}(q_1) | \epsilon - \text{closure}(q_2) = \{q_1, q_2\} = Q_2$

Q_2 переход по b(Q_5)

$Q_5 = \epsilon - \text{closure}(q_1) | \epsilon - \text{closure}(q_3) = \{q_1, q_3\}$

Q_5 переход по a(Q_6)

$Q_6 = \epsilon - \text{closure}(q_1) | \epsilon - \text{closure}(q_2) | \epsilon - \text{closure}(q_4) = \{q_1, q_2, q_4\}$

Q_5 переход по b(Q_1)

$Q_7 = \epsilon - \text{closure}(q_1) = \{q_1\} = Q_1$

Q_6 переход по a(Q_8)

$Q_8 = \epsilon - \text{closure}(q_1) | \epsilon - \text{closure}(q_2) | \epsilon - \text{closure}(q_5) = \{q_1, q_2, q_5\}$

Q_6 переход по b(Q_5)

$Q_9 = \epsilon - \text{closure}(q_1) | \epsilon - \text{closure}(q_3) = \{q_1, q_3\} = Q_5$

Q_8 переход по a(Q_2)

$Q_{10} = \epsilon - \text{closure}(q_1) | \epsilon - \text{closure}(q_2) = \{q_1, q_2\} = Q_2$

Q_8 переход по b(Q_{11})

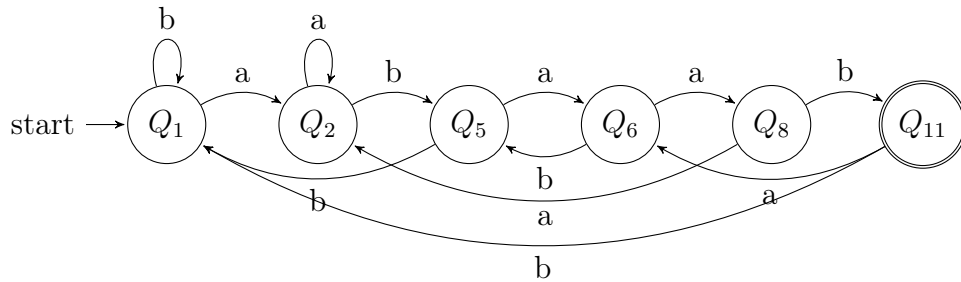
$Q_{11} = \epsilon - \text{closure}(q_1) | \epsilon - \text{closure}(q_3) | \epsilon - \text{closure}(q_6) = \{q_1, q_3, q_6\}$

Q_{11} переход по a(Q_6)

$Q_{12} = \epsilon - \text{closure}(q_1) | \epsilon - \text{closure}(q_2) | \epsilon - \text{closure}(q_4) = \{q_1, q_2, q_4\} = Q_6$

Q_{11} переход по b(Q_1)

$Q_{13} = \epsilon - \text{closure}(q_1) = \{q_1\} = Q_1$



6.

а) Докажем это от противного. Допустим, что ДКА имеет меньше, чем $|w| + 1$ состояний. Тогда рассмотрим цепочку из последовательности состояний, когда мы подадим в наш автомат слово, принадлежащее нашему языку, а точнее на последние $|w| + 1$ состояний. В силу того, что у нас меньше, чем $|w| + 1$ состояний, то по признаку Дирихле в этой цепочке существует состояние, которое встречается 2 раза (состояние X). Рассмотрим промежуток между тем моментом, когда мы встретили это состояние X в 1 раз и когда мы встретили его во второй раз. Если мы выкинем эту часть цепочки, то автомат будет допускать это слово, но при этом его суффикс уже будет другим (проблема может возникнуть, если суффикс не изменится, но в силу произвольности слова, мы всегда можем подобрать такое слово, чтобы он изменился, достаточно лишь (если мы выкидываем цепочку из k символов) позаботиться, чтобы k символов до суффикса не совпадали с первыми k символами суффикса). Ура, мы пришли к противоречию.

б) Давайте сделаем такой автомат, который поддерживает следующие инварианты: во втором состоянии последний символ входа, который мы прочитали - первая буква суффикса, причем последние 2 буквы входа не являются первыми буквами суффикса, в третьем состоянии последние 2 символа входа, которые мы прочитали - две первые буквы суффикса, причем последние 3 буквы входа не являются первыми буквами суффикса, и т.д. На $|w| + 1$ состоянии последние $|w|$ символов входа, которые мы прочитали - $|w|$ первых букв суффикса.

Обозначения: $w[i]$ - i-ая буква суффикса, $W[i]$ - первые i букв суффикса, $\neg w[i]$ - другая буква алфавита, которая не стоит на i-ом месте. $\Theta(i, x)$ (где i - последовательность символов, x - входной символ) - функция, которая возвращает значение, какое максимальное количество символов можно взять из конца новой последовательности символов, чтобы они были первыми буквами суффикса (ясно, что $\Theta(i, w[i]) = k + 1$, где k - длина последовательности i)

Для этого определим δ следующим образом: $\delta(q_i, w[i]) = q_{i+1}$, $\delta(q_i, \neg w[i]) = q_k$, где $k = \Theta(W[i] \cdot \neg w[i]) + 1$

Докажем по индукции, что при такой функции соблюдаются инварианты:

База: при длине слова $n=0$ находимся в 1 состоянии, это соответствует нашему инварианту-потому что 0 символов и не больше соответствуют первым 0 символам из суффикса.

Переход: пусть для n -состояния соблюдаются инварианты. Докажем для $n+1$: мы стоим на i месте, если $x=w[i]$, то мы перейдем на $i+1$ место и так как мы добавили одну букву $w[i]$, то в $i+1$ состоянии последние i символа входа, которые мы прочитали- i первые буквы суффикса, причем последние $i+1$ буквы входа не являются первыми буквами суффикса. Инвариант сохранился.

Если же $x \neq w[i]$, то мы перейдем на то место, где сохранится инвариант в силу определения функции Θ

Инварианты доказаны, в силу этих инвариантов наш алгоритм допускает только слова, оканчивающиеся на нужный суффикс (инвариант допускающего состояния) и все слова, оканчивающиеся на этот суффикс, допустимы (ведь автомат полный и если бы мы находились в состоянии отличном от допустимого, то наш инвариант бы не соблюдался).

7.

Рассмотрим наш ДКА, если в нем меньше, чем 2^n состояний, то найдутся такие две цепочки a, b длины n , что у них отличается хотя бы один символ на i месте и их цепочки заканчиваются на месте q . Тогда, если i равно 1, то q должно быть одновременно допускающим и недопускающим. Если же i больше 1, то рассмотрим состояние q' , в которое мы перейдем из q по $i-1$ нулю. В это состояние попадут наши новые цепочки $a' = a \cdot 00...00$ и $b' = b \cdot 00...00$. Эти цепочки попадут в состояние q' и для одной оно должно быть допускающим, а для другой нет. Противоречие. Следовательно в ДКА хотя бы 2^n состояний.