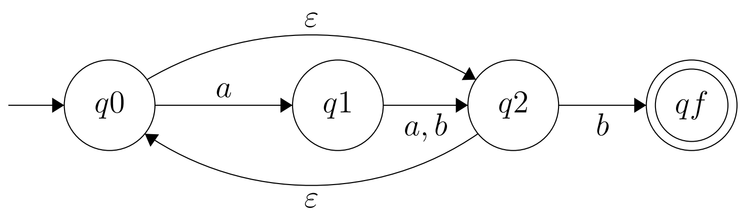
Домашняя работа по ТРЯП №2

Автор - Айвазов Денис из 671 группы

14 сентября 2017

1 НКА по РВ $(a(a|b))^*b$

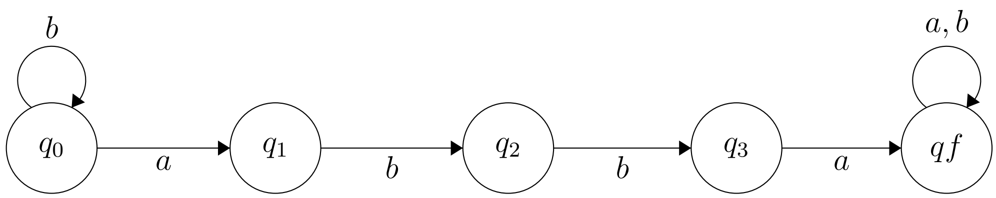


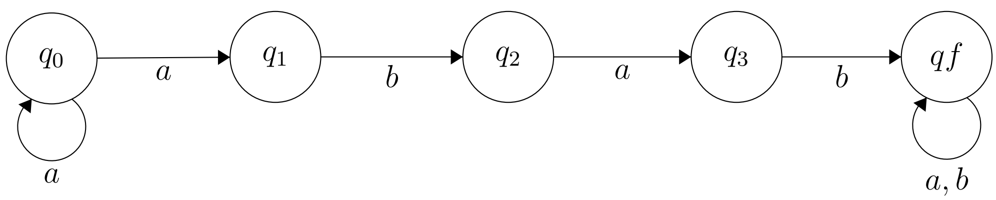
Вот автомат. Докажем, что он принимает язык $L = (a(a|b))^*b$. По индукции по степени выражения $(a(a|b))^*$ докажем, что он принимает язык. База: он принимает слова b,ab, aab и abb.
Шаг индукции: пусть он принимает слова вида $\omega = (a(a|b))^n b$.
Значит если он принял ω, то остановился в q_f. В которое перешёл по b из q2. Т.е. обработав слово вида $(a(a|b))^n$ автомат остановится в q2(или в q0 так как там эпсилон переход). Далее получив на вход $(a(a|b))b$ он пройдёт следующую последовательность $q_2(\varepsilon) \to q_0(a) \to q_1(a|b) \to q_2(a) \to q_f$. Таким образом автомат принял на вход слово вида $(a(a|b))^n(a(a|b))b = (a(a|b))^{n+1}b$. Что и требовалось доказать.
Теперь докажем, что все слова, которые принимаются автоматом имеют вид $(a(a|b))^*b$:
Очевидно, что автомат принимает только слова, оканчивающиеся на b. Получив на вход слово автомат оказывается либо в q0 либо в q2. Если это слово - буква b, то он идёт в qf и слово принято. Получив слово начинающееся с буквы b(длины >1) он ломается, т.к. из qf нет переходов. Если же слово начинается на a, $q_0(a) \to q_1(a|b)q_0[q_f]$. Т.е. слово $a(a|b)b$ будет принято. А если слово вида $a(a|b)a\dots$. То он вернётся по эпсилон переходу в начальное состояние q_0 и начнётся то же самое, пока слово не будет удовлетворять виду $(a(a|b))^*b$

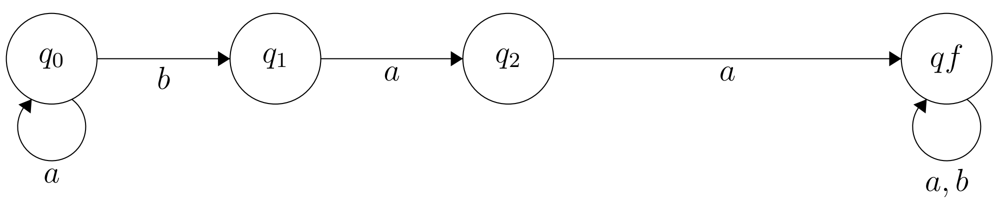
2 $*$ ДКА по НКА вида: $\Sigma^*\omega\Sigma^*$

3 Подстроки abba, abab, baa

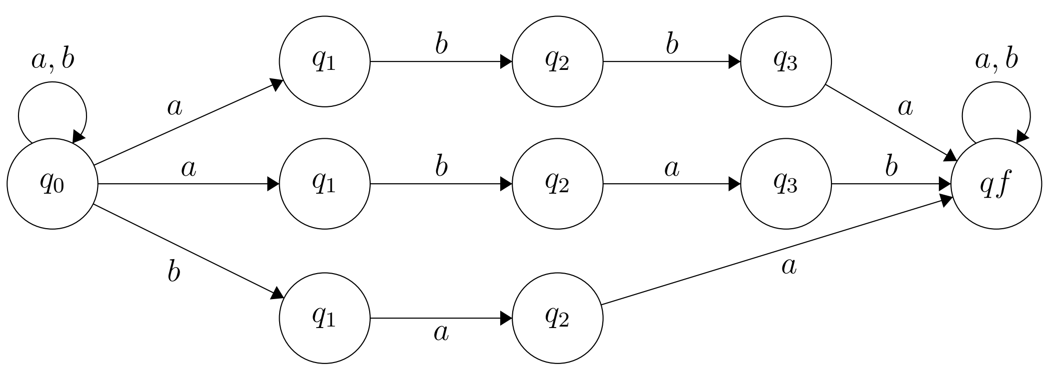
Вот автоматы, распознающие слова, содержащие нужные строки в качестве подслов, поотдельности.







Далее скрепим их с помощью |. Оно работает потому что автомат недетерминированный и будет пытаться проверять по всем веткам одновременно и для достижения q_f достаточно, чтобы он прошёл по какой-то одной ветке.

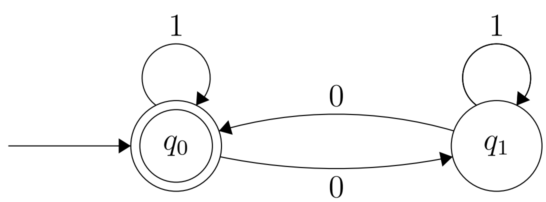


Заметим, что ничего функционально не изменилось, только теперь это автомат, принимающий любое из слов, которые принимали прошлые автомата. Доказательство корректности следует из утвержений из листка с домашним заданием 2.

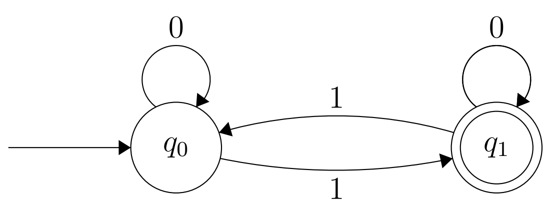
4 Построить ДКА распознающий:

4.1 Язык, все слова которого содержат чётное число нулей

Докажем по индукции по числу нулей. База: пустое слово содержит чётное число(0) нулей. 00 будет обработано так: $q_0(0) \to q_1(0) \to q_0$. По построению каждый ноль меняет состояние с q_0 на q_1 и наоборот. Шаг индукции: пусть он принял слово с чётным числом нулей. Т.е. оказался в q_0. И тогда получив ещё сколько угодно 1 на вход, он останется в q_0 и слово будет принято. Если же 0 оказалось нечётное количество, то автомат окажется в q_1(даже получив кучу 1 после) и пока чётность нулей не изменится, он останется в непринимающем состоянии.



4.2 Язык, все слова которого содержат нечётное число единиц

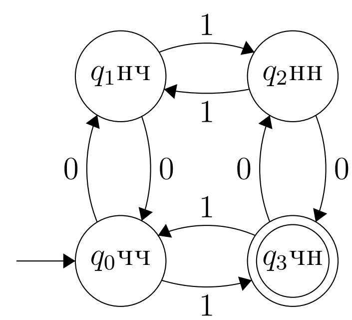


Выглядит и работает абсолютно аналогично. Только теперь 1 и 0 из предыдущего номера поменялись местами. И теперь изменилось положение принимающего состояния. Т.е. приняв нечётное число 1 он остановится в принимающем q_1. в противном случае он не примет слово. Док-во аналогично

4.3 Язык, все слова которого содержат чётное число нулей и нечётное число единиц

Для удобства восприятия в принимающих состояниях напичаны чётности в виде чч,чн,нч,нн(где н= нечётное, ч = чётное. Первая буква говорит о 0, а вторая о 1. Например чн значит, что нулей чётно, а единиц нечётно) Будем смотреть на чётность числа 0 и 1 в обработанном слове при каждом переходе:

q_0 в начале 0 ч и 1 ч
$q_0(0) \to q_1$ - 0 - н, 1 - ч
$q_1(0) \to q_0$ - 0 - ч, 1 - ч
$q_0(1) \to q_3$ - 0 - ч, 1 - н
$q_3(1) \to q_0$ - 0 - ч, 1 - ч
$q_1(1) \to q_2$ - 0 - н, 1 - н
$q_2(1) \to q_1$ - 0 - н, 1 - ч
$q_2(0) \to q_3$ - 0 - ч, 1 - н
$q_3(0) \to q_2$ - 0 - н, 1 - н

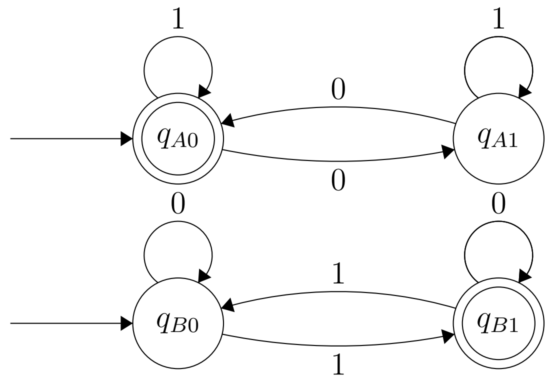


Все вышесказанное следует из переходов и выводов по предудущим задачам. Как мы видим, принимающее состояние имеет описание чн, это значит, что если обработка слова остановилась в нём, то в слове было чётное число 0 и нечётное число 1. Ч.т.д

5 Замена последнего пункта

При алгоритме описанном в листке мы можем рассматривать наш построенный по двум ДКА(A,B) ДКА C в разных случаях как автомат для каждого из A и B в отдельности(игнорируя вторую "координату"состояния). Т.е. если говорить о следующей задаче, можно было бы переходить только по q_i^A и обрабатывать слово для автомата A.
А с учетом того декартового произведения $F_C = F_A \times Q_B \cup Q_A \times F_B$, если нам даётся на вход слово из $L(A)$, то оно по всем состояниям для A приходит в один из $F_A \times Q_B$ и принимается языком. Аналогично со словом из $L(B)$. Из этого и того факта, что $Q_C \supset F_C \supset F_A \times Q_B$ то $L(A) \subset L(C)$ и аналогично $L(B) \subset L(C)$. Но других разрешающих состояний в C нет, значит он не просто принимает, но и распознаёт язык обьединения $L(A) \cup L(B)$

6 Автомат из 4.3 построенный на 4.1 и 4.2



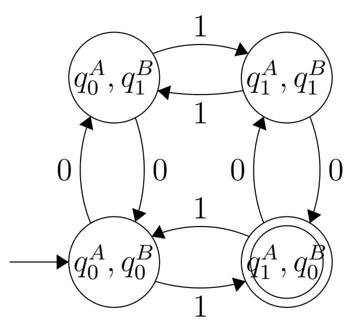
По алгоритму приведённому в листках построим ДКА по двум ДКА. Для простоты нарисуем таблички переходов для двух автоматов A и B.
У нас были автоматы:
$A = (Q_A,\ \Sigma,\ q_0^A,\ \delta_A,\ F_A)$.
$B = (Q_B,\ \Sigma,\ q_0^B,\ \delta_B,\ F_B)$.
Где $\Sigma = \{0, 1\}$; $Q_A = q_0^A, q_1^A$; $Q_B = q_0^B, q_1^B$; $F_A = q_0^A$; $F_B = q_1^B$

δ_A	q_0	q_1	δ_B	q_0	q_1
0	q_1	q_0	0	q_0	q_1
1	q_0	q_1	1	q_1	q_0

Обьединим все это в одну автомат: $C = (Q_C,\ \Sigma,\ q_0^C,\ \delta_C,\ F_C)$., где

$\Sigma = \{0, 1\};\ Q_A = (q_0^A; q_0^B), (q_0^A; q_1^B), (q_1^A; q_0^B), (q_1^A; q_1^B);\ q_0^C = (q_0^A; q_0^B);\ F_c = (q_1^A; q_0^B)$

$\delta_{A\cap B}$	q_0^A, q_0^B	q_0^A, q_1^B	q_1^A, q_0^B	q_1^A, q_1^B
0	q_1^A, q_0^B	q_1^A, q_1^B	q_0^A, q_0^B	q_0^A, q_1^B
1	q_0^A, q_1^B	q_0^A, q_0^B	q_1^A, q_1^B	q_1^A, q_0^B



Таким образом мы построили автомат по алгоритму с листочка. Все состояния и переходы указаны. И мы получили тот же автомат, корректность которого доказали ранее.