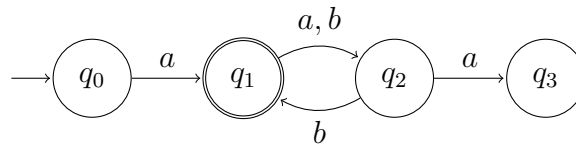


Задание 2

Усвятцов Михаил, группа 176 б

НКА и алгоритмы поиска подстрок

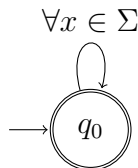
Задача 1. Постройте НКА по регулярному выражению $a((a|b)b)^*$.



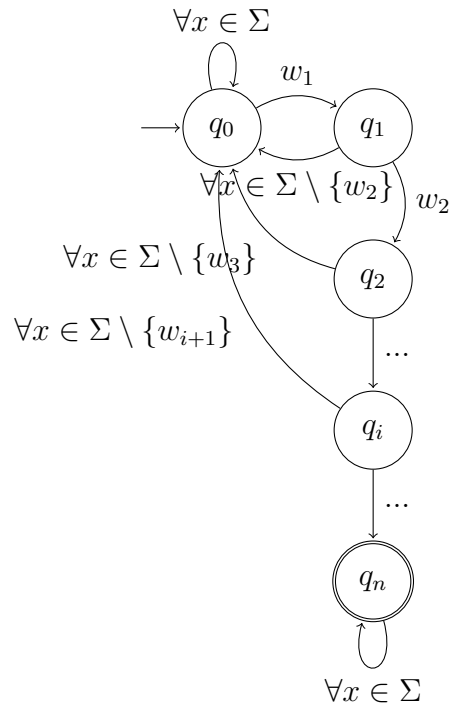
Задача 2*. Докажите, что по НКА данного вида $(\Sigma^*w\Sigma^*)$ можно построить ДКА, число состояний которого не превосходит $|w| + 1$.

Очевидно, что для любой подстроки w длины n можно составить соответствующую часть ДКА с количеством переходов равным n . По одному переходу на букву слова. Состояний у такого автомата будет как раз $n + 1$.

Рассмотрим $w = \varepsilon$. Длина ε слова равна 0. Рассмотрим автомат, принимающий ε слово.

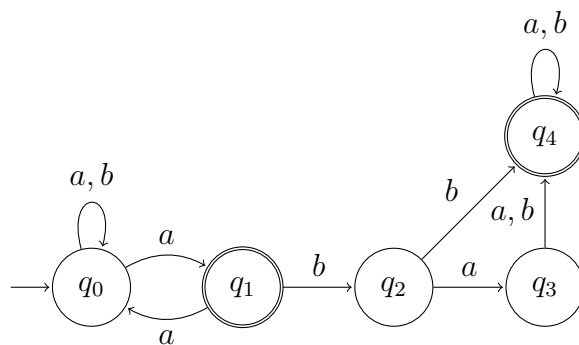


Таким образом, для пустого слова условия задачи выполняются. Предположим, что существует слово w для которого выполняется условие задачи. w_i будем обозначать i -букву слова. $i \in N$. Данный ДКА изображен на рисунке ниже.



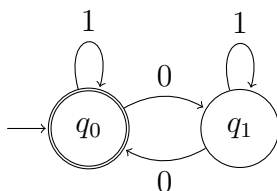
Теперь добавим еще одну букву в слово w . Куда бы мы ни добавили букву к подстроке w , мы увеличим количество состояний ровно на одно. Таким образом для подстроки длиной $n + 1$ количество состояний ДКА стало $n+1 + 1 = n+2$, ЧТД.

Задача 3. Постройте НКА, распознающий слова, в которых есть хотя бы одно из подслов $abab, abb, abaa$.

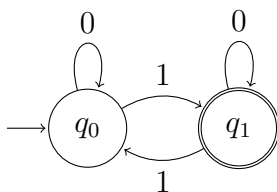


Задача 4. Постройте ДКА, который

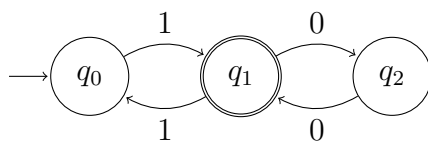
1) распознаёт язык, все слова которого содержат чётное число нулей;



2) распознаёт язык, все слова которого содержат нечётное число единиц;

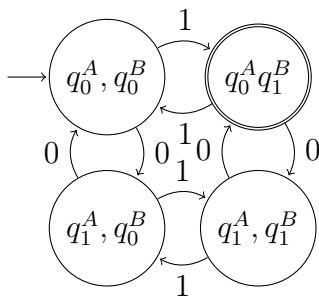


3) распознаёт язык, все слова которого содержат чётное число нулей и нечётное число единиц.



Задача 5. Постройте автомат из задачи 4(3) используя автоматы, построенные в первых двух пунктах и конструкцию произведения.

Обозначим автомат из пункта 1 за А, а автомат из пункта 2 за В. Тогда $Q_C = Q_A \times Q_B = \{q_0^A, q_0^B, q_1^A, q_1^B\}$
 $q_0^C = (q_0^A, q_0^B)$



Задача 6. Как изменить конструкцию произведения, чтобы в её результате автомат \mathcal{C} распознавал язык $L(\mathcal{A}) \setminus L(\mathcal{B})$ – разность языков, распознаваемых ДКА \mathcal{A} и \mathcal{B} ?

- $Q_{\mathcal{C}} = Q_{\mathcal{A}} \times Q_{\mathcal{B}}$;
- $q_0^{\mathcal{C}} = (q_0^{\mathcal{A}}, q_0^{\mathcal{B}})$;
- $\forall \sigma \in \Sigma : \delta_{\mathcal{C}}((q_{\mathcal{A}}, q_{\mathcal{B}}), \sigma) = (\delta_{\mathcal{A}}(q_{\mathcal{A}}, \sigma), q_i \in Q_{\mathcal{B}} \setminus \{\delta_{\mathcal{B}}(q_{\mathcal{B}}, \sigma)\})$;
- $F_{\mathcal{C}} = F_{\mathcal{A}} \times Q_{\mathcal{B}} \setminus Q_{\mathcal{A}} \times F_{\mathcal{B}}$.

Задача 7*. Мы рассматривали алгоритм построения ДКА по РВ, в котором не встречается пустое слово. В случае когда оно встречается, исходное РВ R может быть преобразовано либо в выражение R' , либо в выражение $R' \mid \varepsilon$, где в выражение R' пустое слово уже не входит. Предложите алгоритм, который осуществляет такое преобразование.

Задача 8*. Мы рассматривали алгоритм построения ДКА по РВ, в котором не встречается пустое множество. В случае когда оно встречается, исходное РВ R может быть преобразовано либо в выражение R' , либо в выражение $R' \mid \varepsilon$, причём пустое множество и пустое слово в выражение R' не входят. Предложите алгоритм, который осуществляет такое преобразование.

Указание: Введите новый атрибут в дерево.