



Министерство науки и высшего образования Российской Федерации
Федеральное государственное бюджетное образовательное учреждение
высшего образования
«Московский государственный технический университет
имени Н.Э. Баумана
(национальный исследовательский университет)»
(МГТУ им. Н.Э. Баумана)

ФАКУЛЬТЕТ ИУ «Информатика и системы управления»

КАФЕДРА ИУ-7 «Программное обеспечение ЭВМ и информационные технологии»

ЛАБОРАТОРНАЯ РАБОТА № 1

ПО КУРСУ:

«Конструирование компиляторов»

ПО ТЕМЕ:

«Распознавание цепочек регулярного языка»

Студент ИУ7-21М

_____ *Петров И.А.*

Преподаватель

_____ *Ступников А.А.*

Москва 2025 г.

1. Описание задания

Цель работы: приобретение практических навыков реализации важнейших элементов лексических анализаторов на примере распознавания цепочек регулярного языка.

Задачи работы:

1. Ознакомиться с основными понятиями и определениями, лежащими в основе построения лексических анализаторов.
2. Прояснить связь между регулярным множеством, регулярным выражением, праволинейным языком, конечно-автоматным языком и недетерминированным конечно-автоматным языком.
3. Разработать, тестировать и отладить программу распознавания цепочек регулярного или праволинейного языка в соответствии с предложенным вариантом грамматики.

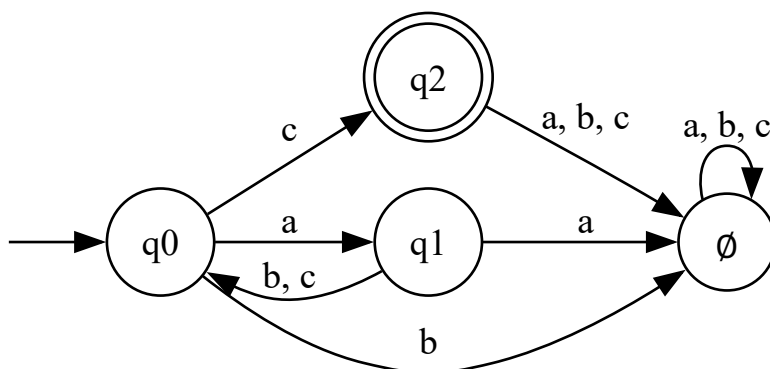
Вариант 7: Напишите программу, которая в качестве входа принимает произвольное регулярное выражение, и выполняет следующие преобразования:

1. Преобразует регулярное выражение непосредственно в ДКА.
2. По ДКА строит эквивалентный ему КА, имеющий наименьшее возможное количество состояний с помощью алгоритма Бржозовского
3. Моделирует минимальный КА для входной цепочки из терминалов исходной грамматики.

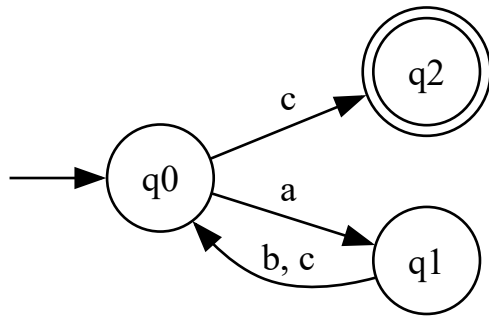
2. Результаты работы программы

1. Введенное регулярное выражение: $(a(b|c))^*c$

Построенный ДКА:



Минимальный КА:



Моделирование для цепочек:

"c": True

"abc": True

"acc": True

"abacabacabc": True

"abacabacacc": True

"abacbc": False

"aabbcc": False

"ab": False

"acb": False

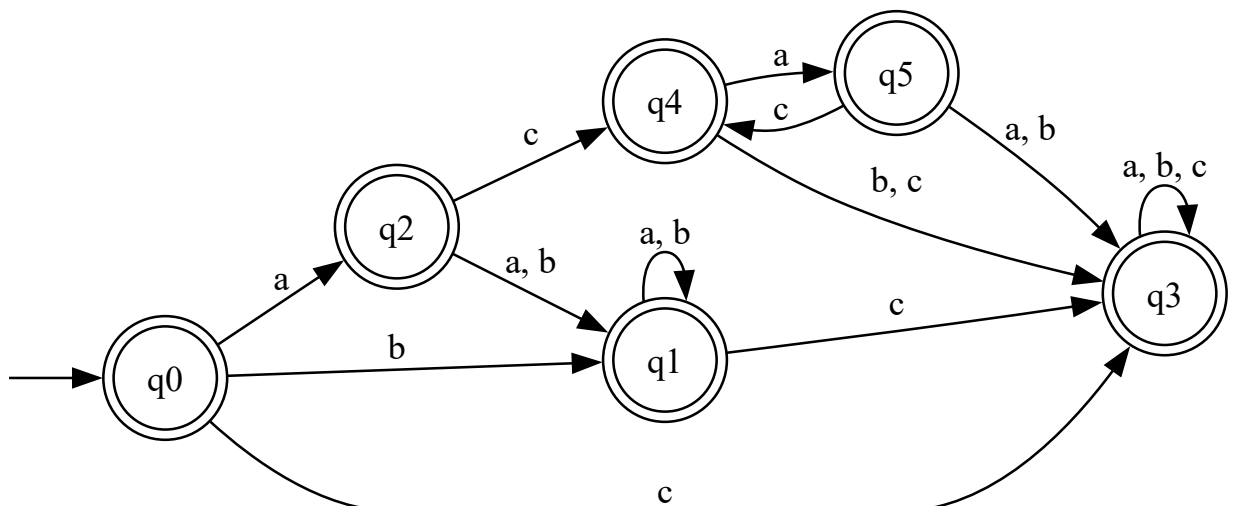
"aac": False

" ": False

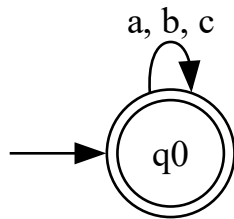
"f": False

2. Введенное регулярное выражение: $(a|b)^*(a|b)|(ac)^*(a|b|c)^*$

Построенный ДКА:



Минимальный КА:



Моделирование для цепочек:

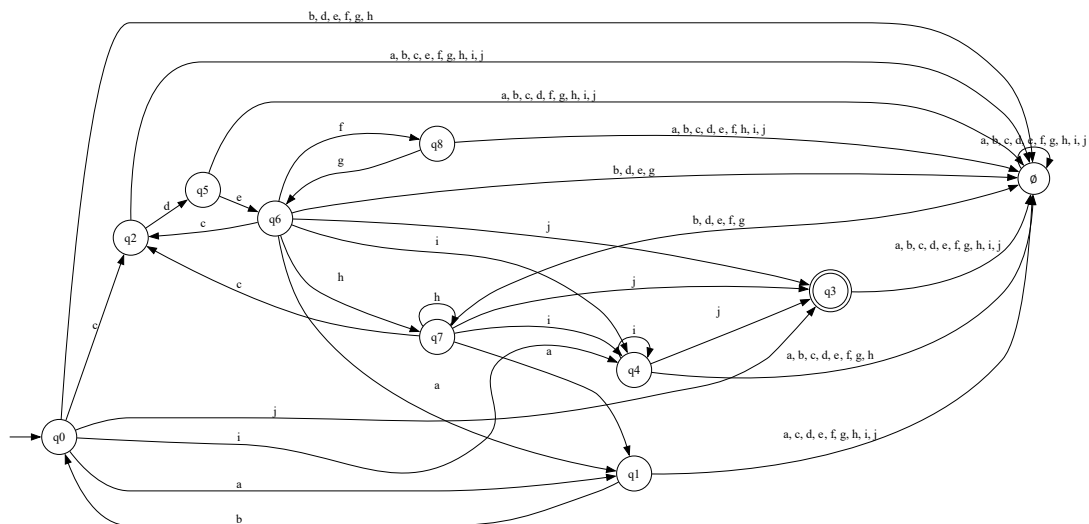
"a": True

"abcaa": True

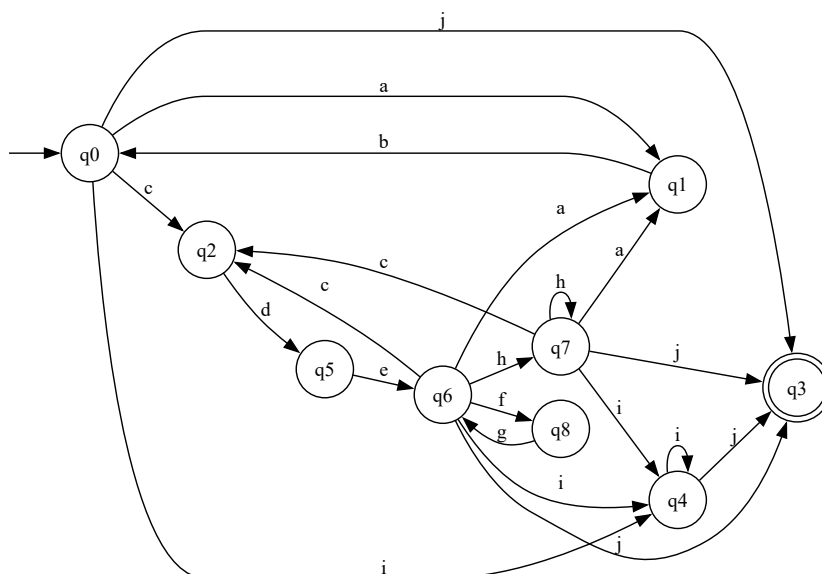
"f": False

3. Введенное регулярное выражение: $(a.b|c.d.(e.(f.g)^*).h^*)^*.i^*.j$

Построенный ДКА:



Минимальный КА:



Моделирование для цепочек:

"j": True

"ij": True

"abj": True

"cdej": True

"cdefgj": True

"cdefgfgj": True

"cdehj": True

"cdefghhhj": True

"ababcdefghiiij": True

"cdefgfgghiiiiij": True

"abcs": False

"cdij": False

"abce": False

"cdhj": False

Заключение

В ходе выполнения лабораторной работы была реализована программа на языке Python, позволяющая распознавать цепочки регулярного языка. В программе было реализовано преобразование регулярного выражения непосредственно в ДКА, по ДКА был построен эквивалентный ему КА, имеющий наименьшее возможное количество состояний, с помощью алгоритма Бржозовского, а также был смоделирован минимальный КА для входной цепочки из терминалов исходной грамматики.

Таким образом, в результате выполнения лабораторной работы были приобретены практические навыки реализации важнейших элементов лексических анализаторов на примере распознавания цепочек регулярного языка.