



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 «Информатика, искусственный интеллект и системы управления»

КАФЕДРА «Программное обеспечение ЭВМ и информационные технологии»

ОТЧЕТ

по лабораторной работе № 1
по курсу «Конструирование компиляторов»
на тему: «Распознавание цепочек регулярного языка»
Вариант № 6

Студент ИУ7-21М
(Группа)

(Подпись, дата)

Миронов Г. А.
(И. О. Фамилия)

Преподаватель

(Подпись, дата)

Ступников А. А.
(И. О. Фамилия)

2024 г.

1 Цель и задачи работы

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, протестировать и отладить программу распознавания цепочек регулярного или праволинейного языка в соответствии с предложенным вариантом грамматики.

2 Выполнение индивидуального задания

2.1 Постановка задачи

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.
3. Моделирует минимальный КА для входной цепочки из терминалов исходной грамматики.

2.2 Примеры работы программы

Регулярное выражение: $(a|b)^*|(a|b)|(ac)^*|(a|b|c)^*$

2.2.1 Консольный интерфейс

Листинг 2.1 – Пример запуска программы. Часть 1

```
$ ./lab-01 -vv '(a|b)^*|(a|b)|(ac)^*|(a|b|c)^*'
DFA of regex:
Q0: 0
F: {0, 1, 2, 3, 4, 5}
Transition table:
      0      1      2      3      4      5
0      a      b      c
1      a,b    c
2      a,b    c
3      a,b,c
4      b,c    a
5      b,a    c
Minimized DFA:
Q0: 0
F: {0}
Transition table:
      0
0      b,c,a
Enter string to check or \q to exit:
```

Листинг 2.2 – Пример запуска программы. Часть 2

```
Enter string to check or \q to exit:
a
cur state (0): check for 'a':
| b -> 0
| c -> 0
| a -> 0
a: true
Enter string to check or \q to exit:
abcaa
cur state (0): check for 'a':
| a -> 0
| b -> 0
| c -> 0
cur state (0): check for 'b':
| c -> 0
| a -> 0
| b -> 0
cur state (0): check for 'c':
| c -> 0
| a -> 0
| b -> 0
cur state (0): check for 'a':
| c -> 0
| a -> 0
| b -> 0
cur state (0): check for 'a':
| c -> 0
| a -> 0
| b -> 0
abcaa: true
Enter string to check or \q to exit:
f
cur state (0): check for 'f':
| a -> 0
| b -> 0
| c -> 0
f: false
Enter string to check or \q to exit:
\q
```

2.2.2 Построенные изображения

На рисунках 2.1–2.3 представлена визуализация работы программы.

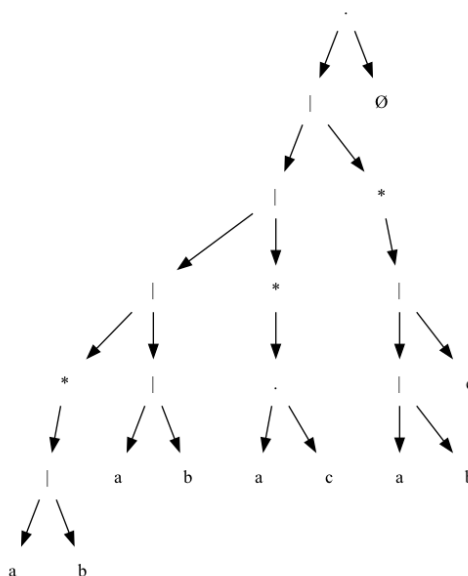


Рисунок 2.1 – Распознанное дерево

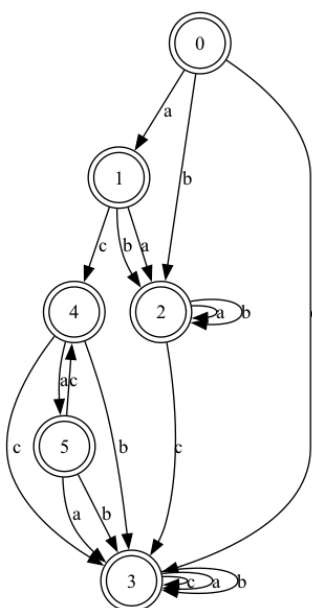


Рисунок 2.2 – Построенный ДКА

2.3 Контрольные вопросы

1. Какие из следующих множеств регулярны? Для тех, которые регулярны, напишите регулярные выражения.

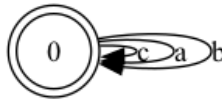


Рисунок 2.3 – Минимизированный ДКА

- (a) Множество цепочек с равным числом нулей и единиц.

Не является регулярным множеством.

- (b) Множество цепочек из $\{0, 1\}^*$ с четным числом нулей и нечетным числом единиц.

Приложил ДКА отдельно

- (c) Множество цепочек из $0, 1^*$, длины которых делятся на 3.

$((0|1)(0|1)(0|1))^*$

- (d) Множество цепочек из $0, 1^*$, не содержащих подцепочки 101.

$(100)^*0^*(1|00|000)^*0^*$

2. Найдите праволинейные грамматики для тех множеств из вопроса 1, которые регулярны.
3. Найдите детерминированные и недетерминированные конечные автоматы для тех множеств из вопроса 1, которые регулярны.
4. Найдите конечный автомат с минимальным числом состояний для языка, определяемого автоматом

$M = (\{A, B, C, D, E\}, \{0, 1\}, d, A, \{E, F\})$,

где функция d задается таблицей

Таблица 2.1 – Переходы в автомате

Состояние	Вход	
	0	1
A	B	C
B	E	F
C	A	A
D	F	E
E	D	F
F	D	E

ПРИЛОЖЕНИЕ А

Ответы на контрольные вопросы