САНКТ-ПЕТЕРБУРГСКИЙ ГОСУДАРСТВЕННЫЙ УНИВЕРСИТЕТ

Факультет прикладной математики - процессов управления

### Полоз Алексей Евгеньевич

Курсовая работа

Разработка и построение конечного автомата для цепочек заданного регулярного языка

2 курс, 231 группа

Руководитель курсовой работы

А.В. Матросов

2018 г.

Санкт-Петербург

2018

# Постановка задачи

1. Разработать распознающий КА для цепочек заданного регулярного языка и использовать его в программе поиска цепочек этого языка во входной последовательности символов.
2. Построить таблицу переходов распознающего КА.

Строка символов **a, b, +, -**, начинающаяся с префикса **+** и заканчивающаяся суффиксом **-**, между которыми располагается последовательность символов **a, b**, в которой после каждого символа a следует последовательность символов **b**, заключённая в символы **+** и **-**, например **+-** или **+a+b-a+bb--**.

# Разработка автомата

По поставленной задаче составлено регулярное выражение: **+(a+b\*-)\*-**.

Разработан ε-НКА:

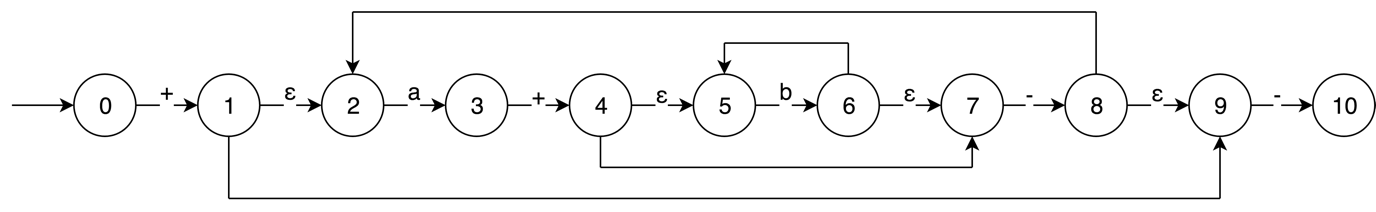
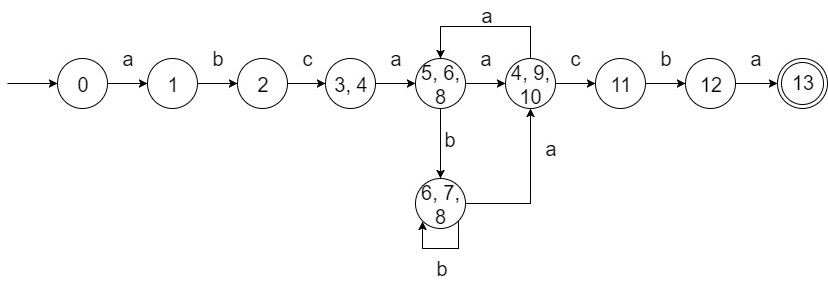


рис. 2.1

Построена таблица его переходов:

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
|  | + | - | a | b |
| ➝{0} | {1, 2} | Ø | Ø | Ø |
| {1, 2} | Ø | {9} | {2, 3} | Ø |
| {3} | {4, 5} | Ø | Ø | Ø |
| {4, 5} | Ø | {7} | Ø | {5, 6, 7} |
| {5, 6, 8} | {4, 9, 10} | {6, 7, 8} | Ø | Ø |
| {11} | Ø | {12} | Ø | Ø |
| {4, 9, 10} | {5, 6, 8} | Ø | {11} | Ø |
| {6, 7, 8} | {4, 9, 10} | {6, 7, 8} | Ø | Ø |
| {12} | {13} | Ø | Ø | Ø |
| \*{13} | Ø | Ø | Ø | Ø |

И получен следующий граф:



# Описание работы программы

Программа представляет собой реализация переходов конечного автомата в различные состояния. Переменная *\_currState* является хранителем текущего состояния. При обработке каждого нового символа *\_currentSymbol* в строке программа проверяет соответствие текущего состояния и нового символа и при возможности переходит в следующее состояние. Иначе происходит сброс состояния. При запуске программа считывает строку из файла, в которой будет производиться поиск подстрок, удовлетворяющих регулярному выражению. Если подстрока будет найдена, она будет выведена в консоль, если подстрок несколько - каждая будет выведена с новой строки. Если подстроки не обнаружены, программа не выводит ничего.

# Тестирование

|  |  |
| --- | --- |
| Ввод в программу | Вывод программы |
| abcaacba | abcaacba |
| **abcabaabbbbbaabacba**bccba | abcabaabbbbbaabacba |
| abcababacbaa**abcabaabbacba** | abcabaabbacba |
| **abcaacba**bcabbbacbaa**abcabacba** | abcaacba  abcabbbacba  abcabacba |
| **abcaacba**abcaacba**abcaacba** | abcaacba  abcaacba  abcaacba |
| abcccc**abcabbacba**bccb**abcaacba** | abcabbacba  abcaacba |
| abccba |  |
| abccccccccccccccccccccccbaaaaabbbbb |  |
| abcabaabc |  |
| qweqrrrqewq |  |

# Код программы

u((({aa+})?)(({bb+})?)(a?)(b?))\*u

u((({aa+})?)(({bb+})?)(a?b)\*(b?a))\*u