

МИНОБРНАУКИ РОССИИ
САНКТ-ПЕТЕРБУРГСКИЙ ГОСУДАРСТВЕННЫЙ
ЭЛЕКТРОТЕХНИЧЕСКИЙ УНИВЕРСИТЕТ
«ЛЭТИ» ИМ. В.И. УЛЬЯНОВА (ЛЕНИНА)
Кафедра МО ЭВМ

ОТЧЕТ
по лабораторной работе №5
по дисциплине «Построение и анализ алгоритмов»
Тема: Алгоритм Ахо-Корасик

Студентка гр. 7304

Юреть Е.А.

Преподаватель

Филатов А.Ю.

Санкт-Петербург

2019

Цель работы.

Решение задачи точного поиска набора образцов и для одного образца с джокером.

Описание алгоритма.

В процессе работы алгоритма Ахо-Корасика используется структура данных – вершина бора и построения по нему конечного детерминированного автомата. Строится бор последовательным добавлением исходных строк. Изначально есть 1 вершина, корень - пустая строка. Добавление строки происходит так: начиная в корне, двигаемся по дереву, выбирая каждый раз ребро, соответствующее очередной букве строки. Если такого ребра нет, то мы создаем его вместе с вершиной. Так как процесс добавления строки может остановиться во внутренней вершине, то для каждой строки будем дополнительно хранить признак того является она строкой из условия или нет. Далее, строим конечный детерминированный автомат. Состояние автомата — это какая-то вершина бора. Переход из состояний осуществляется по 2 параметрам — текущей вершине v и символу ch по которому нам надо сдвинуться из этой вершины. Назовем суффиксной ссылкой вершины v указатель на вершину u , такую что строка u — наибольший собственный суффикс строки v , или, если такой вершины нет в боре, то указатель на корень. В частности, ссылка из корня ведет в него же. Если из текущей вершины есть ребро с символом v , то пройдем по нему, в обратном случае пройдем по суффиксной ссылке и запустимся рекурсивно от новой вершины. Алгоритм завершится, когда мы дойдем до конца строки. При поиске с джокером строка-шаблон разделяется по символу-джокеру на строки, каждая из которых добавляется в бор.

Описание функций и структур

- struct Bohr_v - Реализация Бора
- void ini() - Инициализация Бора
- void add(const std::string & s) - добавление образца в бор
- int getAutoMove(int v, int ch) - вычисление перехода от вершины v по ребру ch.
- int getSuff(int v) - получение суффиксальной ссылки
- void out(int v, int i, std::vector<int> & cnt, const std::vector<int> & len) – проверка на подстроку
- void out_2(const std::vector<int> & cnt, int t_size) – вывод ответа
- void find(const std::string & s, std::vector<int> & cnt, const std::vector<int> & len) - поиск по бору

Тестирование.

```
CCCA
1
CC
1 1
2 1
```

- Рисунок 1. 5_1. Пример №1.

```
ASDFGAS
3
AS
GO
GAS
1 1
5 3
6 1
```

- Рисунок 2. 5_1. Пример №2.

```
ASDSDFG
3
SD
DS
SDSD
2 1
3 2
2 3
4 1
```

- Рисунок 3. 5_1. Пример №3.

```
ACT
A$
$
1
```

- Рисунок 5_2. Пример №1.

```
ACGCACACG
$AC
$
4
6
```

- Рисунок 5. 5_2. Пример №2.

```
ACGCACACG
$$C
$
2
4
6
```

- Рисунок 6. 5_2. Пример №3.

Вывод.

В ходе выполнения лабораторной работы был реализован алгоритм, решающий задачу точного поиска набора образцов с помощью алгоритма Ахо-Корасик. В процессе выполнения работы была решена проблема построения суффиксальных ссылок и проблема поиска конечных состояний автомата, построенного из бора.