

## **Документ продукта по учебной практике**

### **Спецификация**

#### **Задание:**

Задачей проекта является создание интерактивного приложения на языке Kotlin, которое отображает работу алгоритма Ахо-Корасик и позволяет взаимодействовать с ним, используя графический интерфейс.

#### **Работа программы:**

- 1) Приложение открывает окно с двумя полями для ввода текста, в которые пользователь вводит текст в первое поле и набор шаблонов во второе поле и по нажатию клавиши Enter или кнопки на интерфейсе “отправить” программа считывает данные(рис. 2).
- 2) Строит по ним бор в терминах алгоритма Ахо-Корасик, после чего выводит на экран изображение бора слева и конечного автомата справа(рис. 3).
- 3) Через дополнительное поле вводятся данные для постройки нового состояния по форме(“название состояния”->”символ перехода”)(пример текста для добавления ребра: abc->c)(рис.4) и по нажатию клавиши Enter или кнопки на интерфейсе “отправить” программа считывает данные, после чего в бор добавляется ребро из первого состояния в новое, автомат перестраивается по новому бору, приложение анализирует изменённый бор и по нему собирает набор шаблонов в зависимости от данных введённых пользователем(рис. 5). В случае если переход по символу уже имеется то ничего не произойдёт.
- 4) Через второе дополнительное окно вводится название состояние для которого нужно изменить статус с конечного на промежуточное и наоборот, после ввода и нажатия клавиши Enter или кнопки на интерфейсе “отправить”, бор изменится, а автомат перестроится

#### **Описание пользовательского интерфейса:**

При запуске приложение имеет два поля для ввода текста(рис. 1), в одно вписывается текст, в другое набор шаблонов разделённых символом “#”(пример набора шаблонов: #ab#ba#aba) на каждом поле имеется интерфейсная кнопка “отправить” по которой пользователь подтверждает введённые данные(рис. 2).

После подтверждения данных на экран выводится два графа представляющих собой бор и конечный автомат, который и является графическим отображением бора с выводом всех вхождений шаблонов в текст по принципу ("номер шаблона" "позиция вхождения") промежуточные значения имеют черную обводку, конечные значения имеют красную обводку, из вершин исходят направленные рёбра. Рёбра зелёного цвета служат обозначением конечных ссылок, а рёбра синего цвета -- суффиксных ссылок (рис. 3).

В окне с графом также имеются два поля для ввода текста, благодаря одному можно добавлять рёбра по форме ("название состояния" -> "символ перехода") (рис. 4 и 5), по второму можно переключать статус состояния с конечного на промежуточное. Для подтверждения данных введенных пользователем можно нажать кнопку Enter, так же в каждом поле имеется кнопка "отправить" (рис. 6 и 7)



The image shows a software window titled "Step 1 начало работы". Inside the window, there is a large red rectangular border. Within this border, there are two text input fields. The first field contains the placeholder text "Введите текст". The second field contains the placeholder text "Введите шаблоны через #".

Рис. 1 – примерная иллюстрация открывшегося приложения

## Step 2 ввод данных

abab

#a#ba

Рис. 2 – ввод данных для построения бора и нахождения подстрок в строке

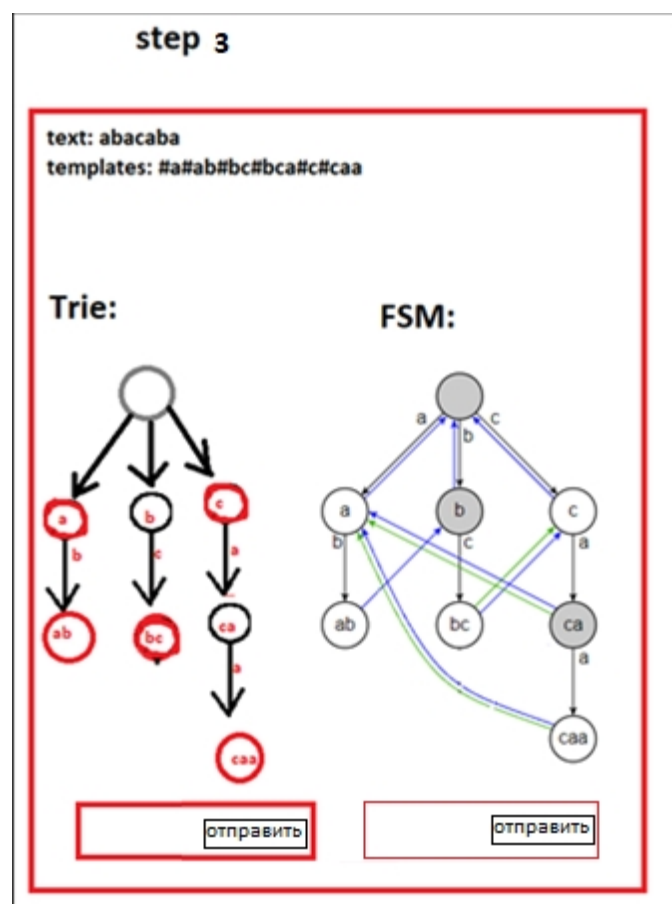


Рис. 3 – иллюстрация вывода

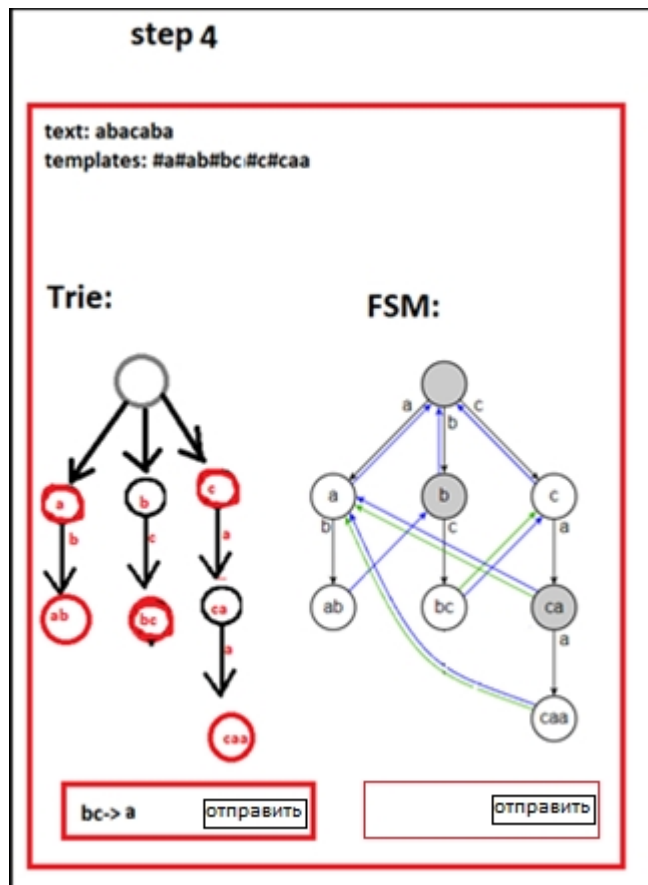


Рис. 4 – иллюстрация записи добавления вершины в бор

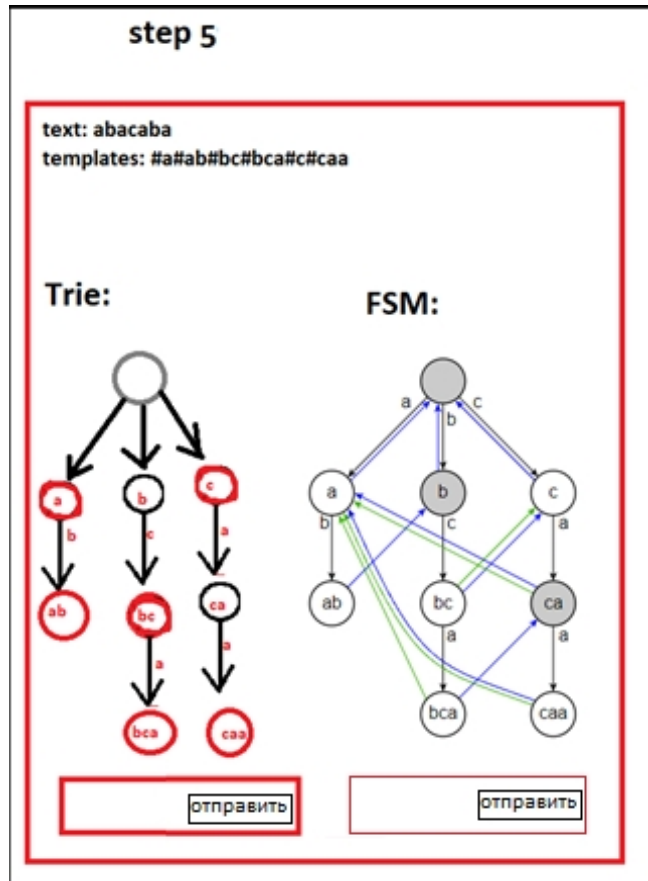


Рис. 5 – иллюстрация изменённого автомата и бора после добавления ребра

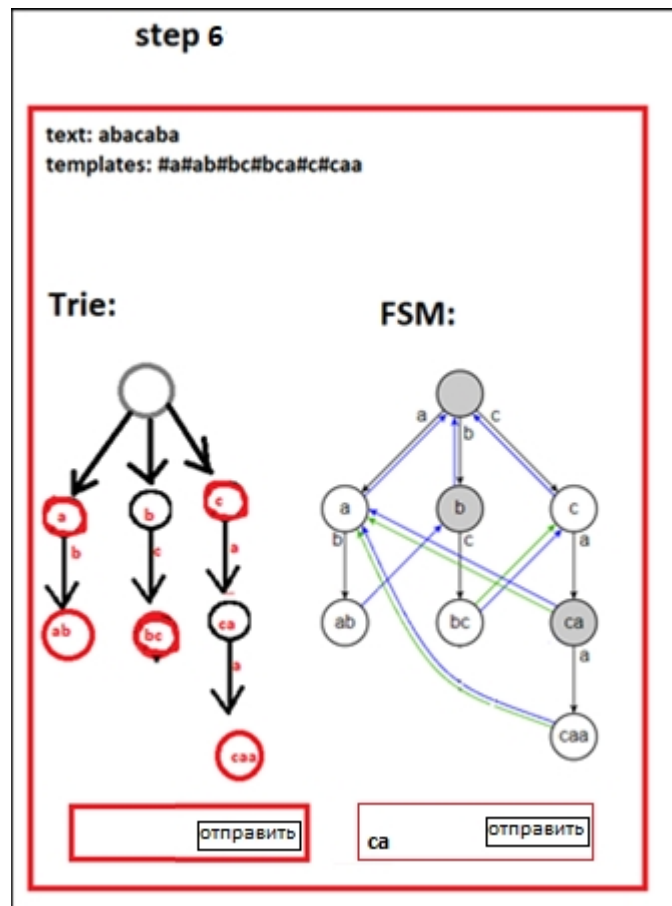


Рис. 6 – иллюстрация записи изменения статуса состояния

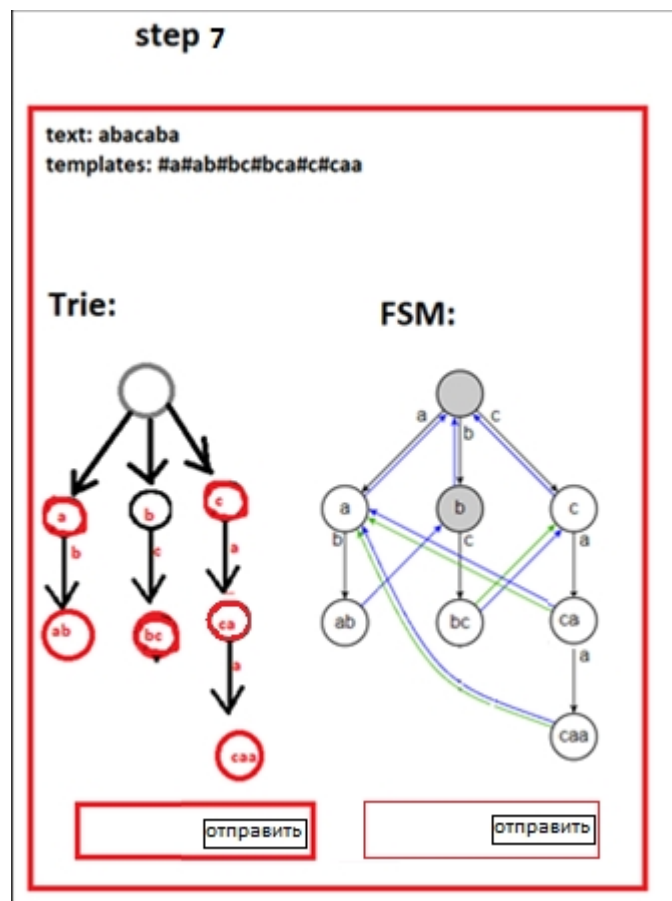


Рис. 7 – иллюстрация изменения бора и автомата

### **Формат входных и выходных данных**

Вход: строка представляющая собой текст в первом окне, серия строк разделённых разделительным знаком “#” во втором окне.

Вывод: Все вхождения шаблонов в текст по принципу ("номер шаблона" "позиция вхождения"), а также рисунок конечного автомата по заданным данным.

### **План разработки**

6 июля: согласование спецификации и плана разработки.

7 июля: создание прототипа программы(приложения, демонстрирующего интерфейс, но не реализующее основные функции);

9 июля: создание первой версии программы с возможностью добавлять переход по состоянию

10 июля: создание второй версии программы с добавленной возможностью удалять переходы по состоянию и возможностью удалять сами состояния

12 июля: сдача финальной версии с отчётом

### **Разделение обязанностей в группе,**

**Голов – общение с преподавателем, оформление документов, алгоритмическая часть**

**Токун – Работа с графическим интерфейсом.**