Министерство науки и высшего образования Российской Федерации

Федеральное государственное автономное образовательное учреждение высшего образования

**«Пермский национальный исследовательский политехнический университет»**

Электротехнический факультет

Кафедра **«**Информационные технологии и автоматизированные системы**»**

направление подготовки: 09.03.04 – **«**Программная инженерия»

**Лабораторная работа №1.**

**“Алгоритмы Маркова и Машина Тьюринга. Решение задач”**

Выполнил студент гр. РИС-24-2б

Бартов Игорь Сергеевич

Проверил:

Доц. каф. ИТАС

Ольга Андреевна Полякова

(оценка) (подпись)

(дата)

г. Пермь, 2024

**Алгоритм Маркова**

1. Дана задача: A={a,b}. Удалить из непустого слова P его первый символ. Пустое слово не менять

К примеру, у нас есть строка: **"abba"**. Наша цель — удалить из неё первый символ.

### Описание решения:

1. Добавляем в начало слова символ, не входящий в массив A
2. Заменяем комбинацию символов терминальной подстановкой в форме \*A, где \* - любой символ, не входящий в массив, A – массив символов.
3. Чтобы учесть, что слово может быть пустое, заменяем “\*” терминальной подстановкой.

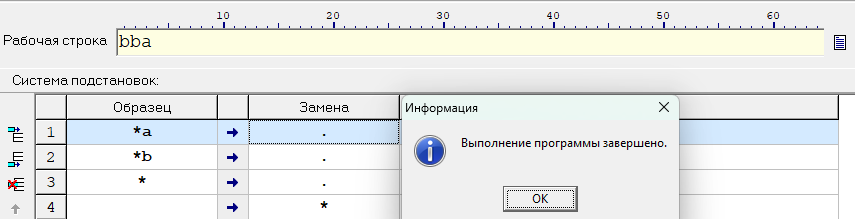
Чтобы проверить, что символ “\*” ещё не содержится в слове, мы расставим правила так, чтобы алгоритм проверил наличие символа, а потом, если он отсутствует, его бы добавил.

**Система подстановок:**

1. **\*a |**→  — если первый символ строки "\*a", программа завершает выполнение, удалив этот символ.
2. **\*b |**→  — если первый символ строки "\*b", программа завершает выполнение, удалив этот символ.
3. **\* |**→ — если первый символ строки "\*", программа завершает выполнение, удалив этот символ.
4. → \* — программа добавляет “\*’ в начало слова

**Выполнение алгоритма на примере:**

1. Проверяем наличие символа “\*” (Правила 1,2,3) в строке и не находим его вхождение: abba
2. По правилу 4 добавляем символ “\*”: \*abba
3. Заменяем “\*a” на терминальную подстановку (Правило 1) и получаем необходимый результат: “bba”



Задача решена

1. Дана задача: A = {a,b,c,d}. В слове P требуется удалить все вхождения символа "c", а затем заменить первое вхождение подслова bb на ddd

Требуется: удалить все вхождения символа "c" из слова P, а затем заменить первое вхождение подслова "bb" на "ddd".

Рассмотрим строку: **abbcabbca**. Наша цель — сначала удалить все "c", а затем заменить первое вхождение "bb" на "ddd".

**Описание решения:**

1. **Удаляем символ “c”**
2. Заменяем "bb" на "ddd": после удаления всех символов "c" нужно заменить первое вхождение "bb" на "ddd". Мы используем терминальную подстановку, чтобы прекратить выполнение после этой замены.

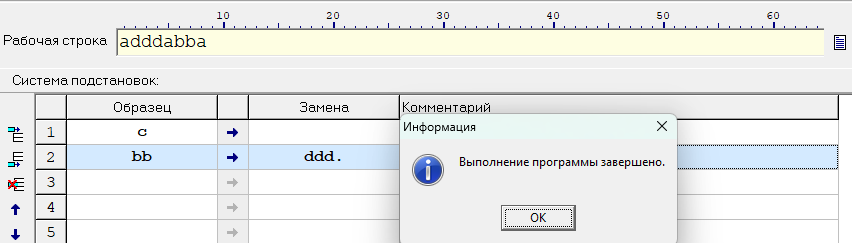
**Система подстановок:**

1. c → (заменяем "c" на пустой символ)

2. bb → ddd. (терминальная замена первого вхождения "bb" на "ddd")

**Выполнение алгоритма на примере:**

1. Начальная строка: **abbcabbca**
2. Применяем первое правило (удаляем все "c"): abbcabbca → abbbabba
3. Применяем второе правило (заменяем первое вхождение "bb" на "ddd"):abbbabba → adddabba



Задача решена

1. Дана задача A={a,b}. Требуется приписать символ "a" к концу слова.

Например, для строки "bbab" результатом будет "bbaba".

### Описание решения:

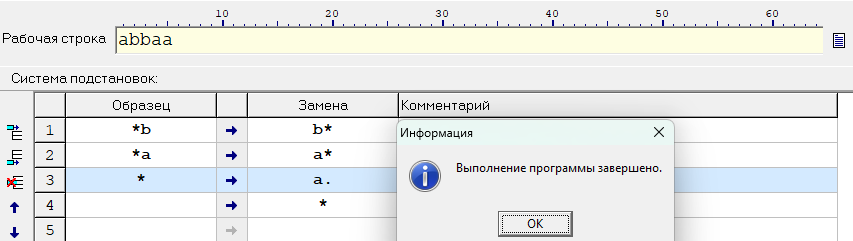
1. Добавим в конец строки символ, не входящий в множество A (например, "\*").
2. Переместим этот символ вправо, пока он не достигнет конца строки.
3. Заменим символ "\*" на "a" терминальной подстановкой.
4. Удалим временный символ "\*".

### Система подстановок:

1. \*b → b\* — если строка содержит комбинацию "b", перемещаем "" вправо.
2. \*a → a\* — если строка содержит комбинацию "a", перемещаем "" вправо.
3. \* → a. — если строка заканчивается на "\*", заменяем его на "a" (терминальная подстановка).
4. → \* — добавляем символ "\*" в конец строки, если он отсутствует.

### Выполнение алгоритма на примере "abba":

1. Проверяем наличие символа "\*". Поскольку его нет, добавляем его в конец строки по правилу 4, получаем: abba\*.
2. Применяем правила 1 и 2:
   * abb\*a → abba\* — "\*" перемещается вправо.
   * ab\*ba → abba\* — "\*" снова перемещается вправо.
   * a\*bb → abba\* — "\*" доходит до конца строки.
3. Применяем правило 3:
   * abba\* → abbaa — заменяем "\*" на "a".
4. Алгоритм завершён. Итоговая строка: "abbaa".



Задача решена