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  
«Нормальные алгорифмы Маркова»**

Выполнил студент гр. РИС-24-2б

Челпанова А. А.

Проверил:

Доц. Каф. ИТАС

Ольга Андреевна Полякова

(оценка) (подпись)

(дата)

г. Пермь, 2024

**Цель работы**

Цель данной работы заключается в реализации нормальных алгорифмов Маркова для решения 3-х задач с использованием среды имитации этой машины.

**Постановка задач**

Задача №1: А={a,b}. Удалить из непустого слова Р его первый символ. Пустое слово не менять.

Задача №2: A={a,b,c,d}. В слове Р требуется удалить все вхождения с, а затем заменить первое вхождение подслова bb на ddd.

Задача №3: A={a,b}. Требуется приписать символ а к концу слова Р.

**Анализ нормальных алгорифмов Маркова.**

Модель алгоритмов Маркова преобразует исходные слова с помощью заданных правил, которые могут заменять часть исходной строки.

Кроме правил в алгоритмах Маркова используются приоритеты и применения этих правил.

1. Правило находит первое своё вхождение в исходной строке и заменяет его на правую часть правила.
2. Правила выполняются по порядку их номеров.
3. Если правило не применимо к исходной строке, то алгоритм переходит к проверке следующего по порядку правилу.
4. Если правило выполнено, то алгоритм возвращается к проверке правил слова, начиная с начала списка.
5. Алгоритм завершается, если ни одно правило неприменимо.

Подстановки и примечания:

1. Нетерминальные.
2. Терминальные – говорят о принудительном завершении программы после выполнения замены.

Если правая часть правила пустая, то левое вхождение в строку исчезает из слова.

Если в левой части правила стоит пустая строка, то в начало слова добавляется правая часть этого правила.

**Задача 1.**

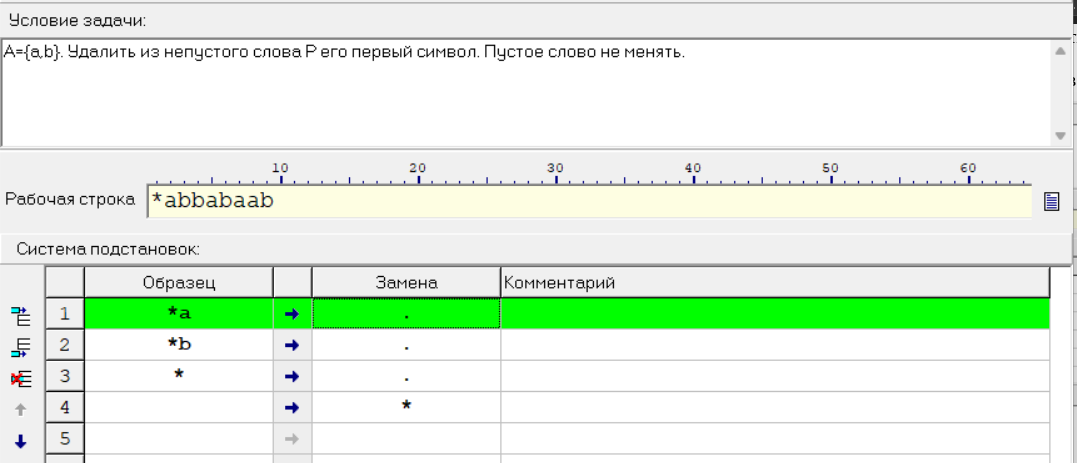
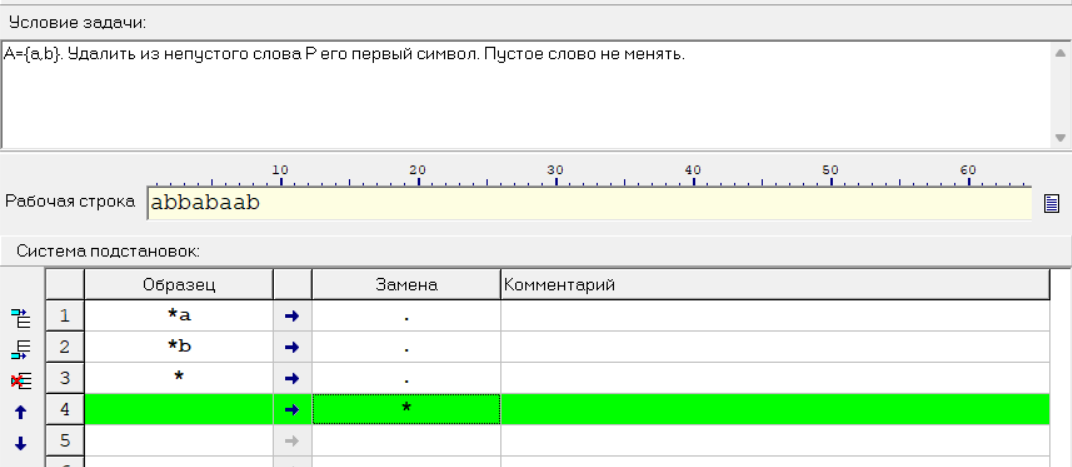
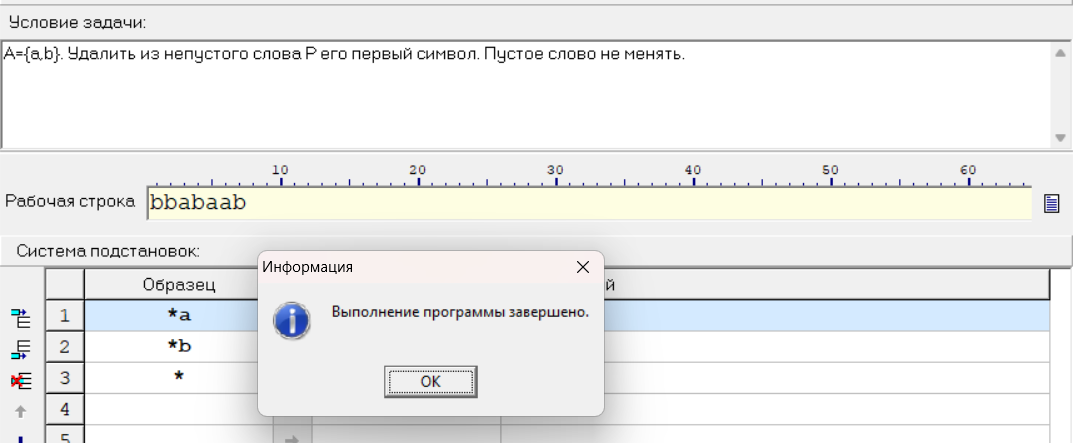
**Последовательность выполнения шагов в соответствии с правилами и визуализацией:**

Исходная строка – abbabaab

Правила:

|  |  |
| --- | --- |
| 1 | \*a|→. |
| 2 | \*b|→. |
| 3 | \*|→. |
| 4 | →\* |

Алгоритм решения:

1. В начало слова добавляется \*;
2. Первое вхождение \*а или \*b удаляется, алгоритм завершается; 
3. Если слово пустое, то удаляется \*, алгоритм завершается.

**Задача 2.**

**Последовательность выполнения шагов в соответствии с правилами и визуализацией:**

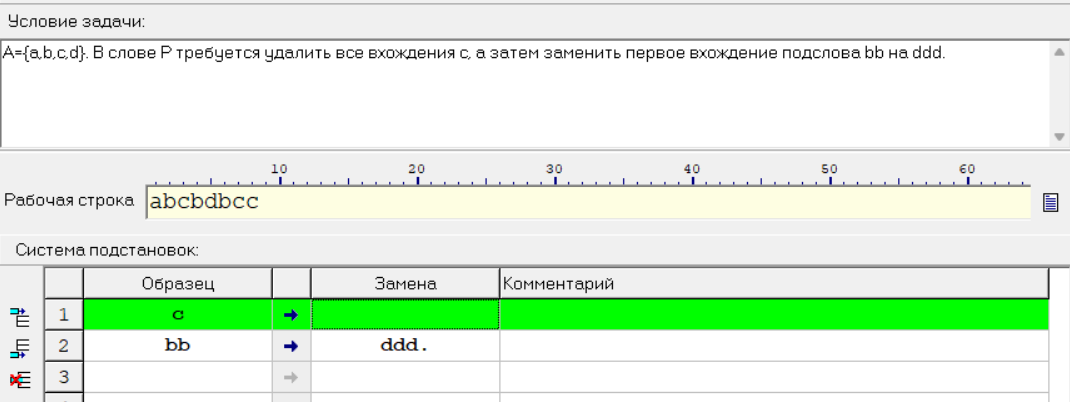
Исходная строка – abcbdbcc

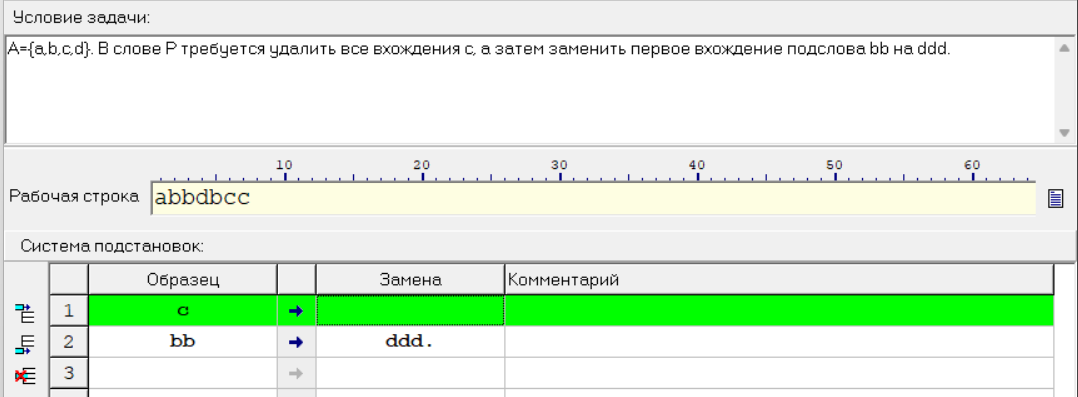
Правила:

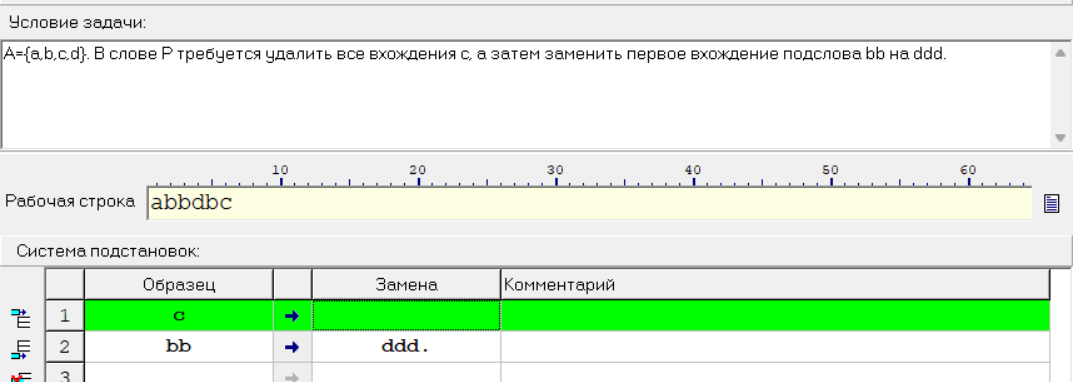
|  |  |
| --- | --- |
| 1 | c→ |
| 2 | bb|→ddd. |

Алгоритм решения:

1. Все вхождения с удаляются;

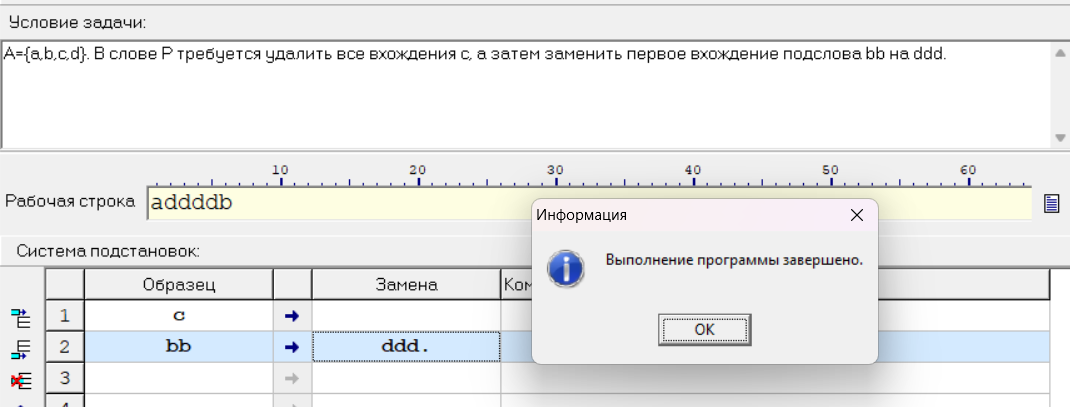






1. Первое вхождение bb заменяется на ddd, алгоритм завершается.





**Задача 3.**

**Последовательность выполнения шагов в соответствии с правилами и визуализацией:**

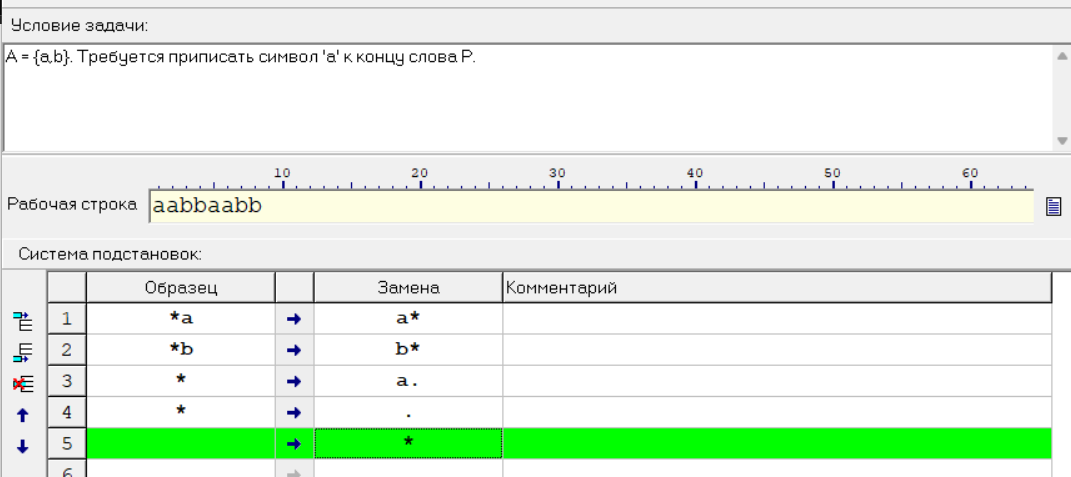
Исходная строка – aabbaabb

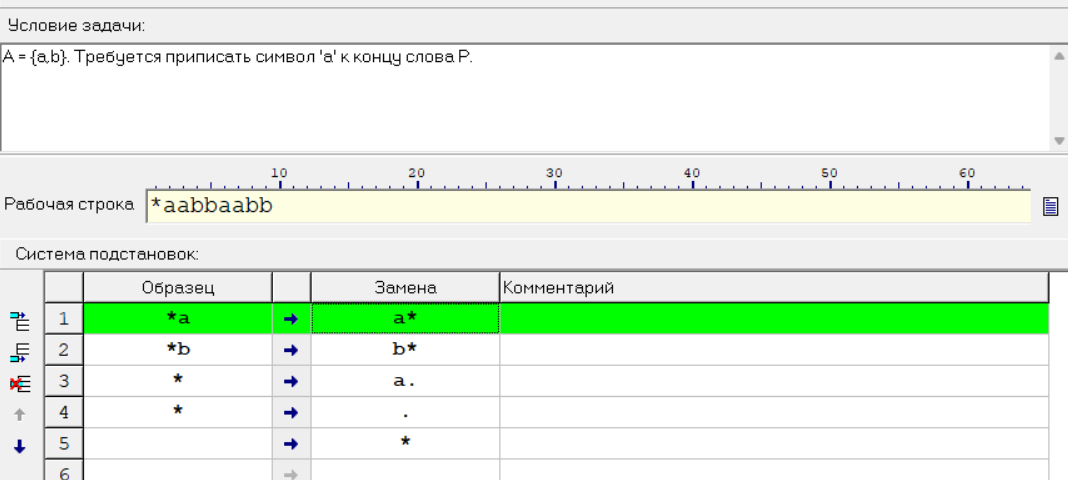
Правила:

|  |  |
| --- | --- |
| 1 | \*a→a\* |
| 2 | \*b→b\* |
| 3 | \*→a. |
| 4 | \*→. |
| 5 | |→\* |

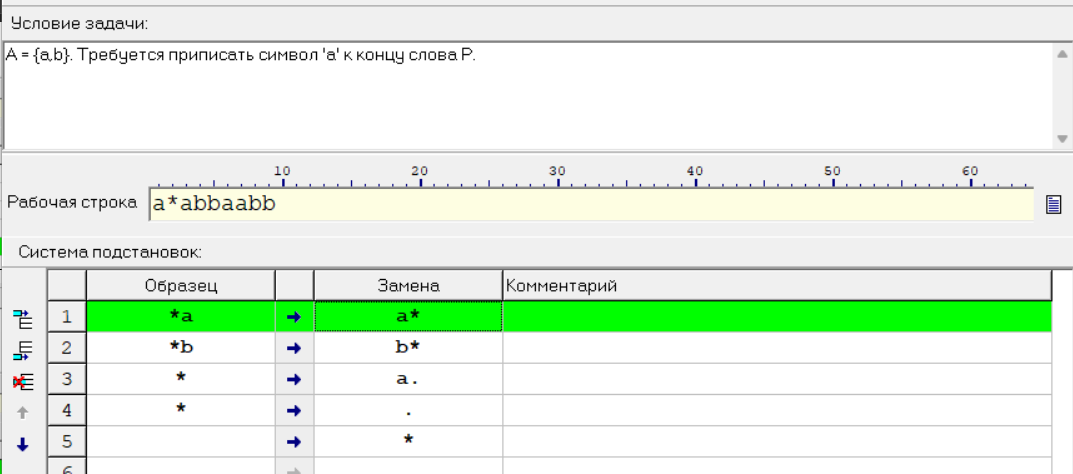
Алгоритм решения:

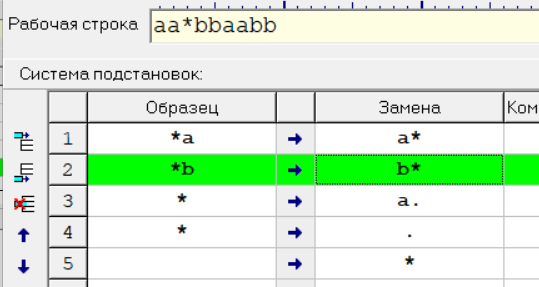
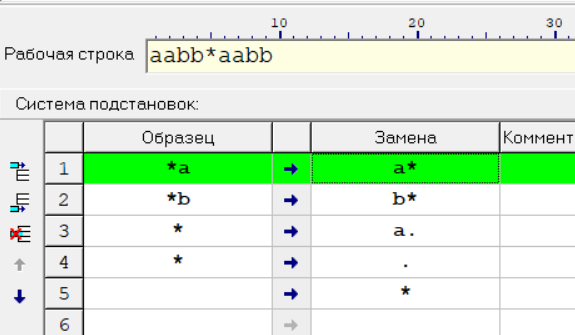
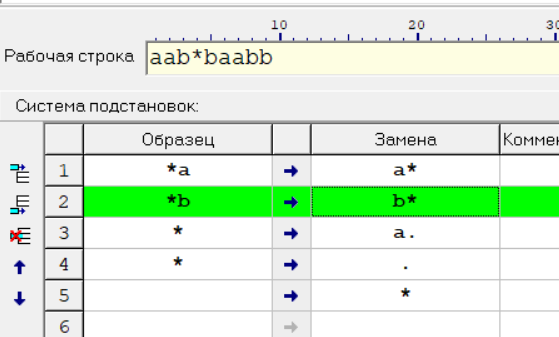
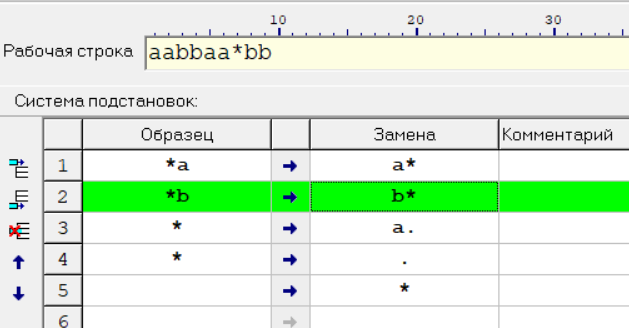
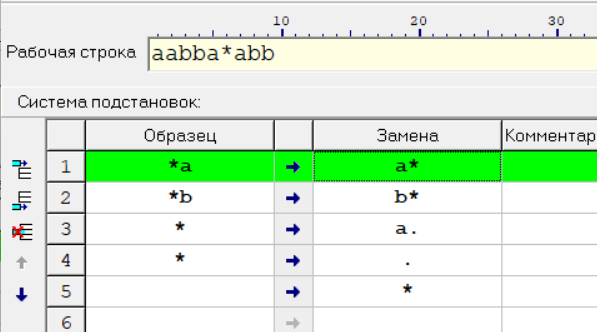
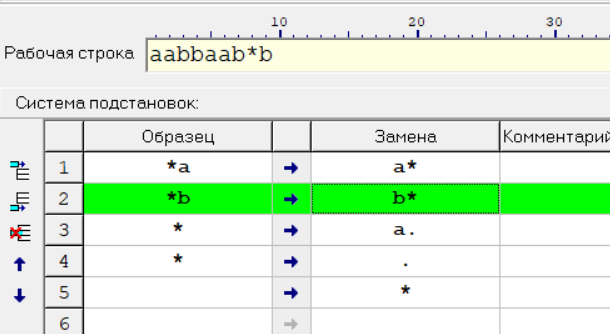
1. К началу слова добавляется \*;



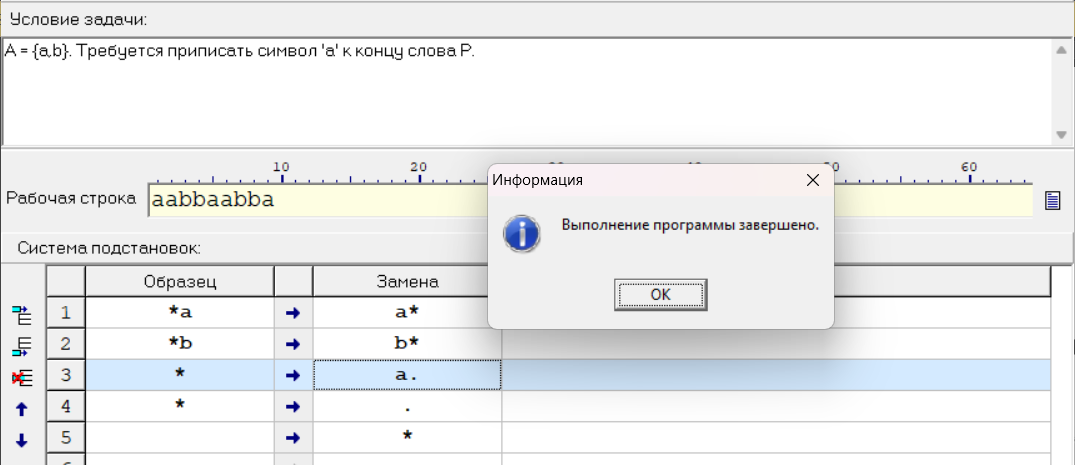
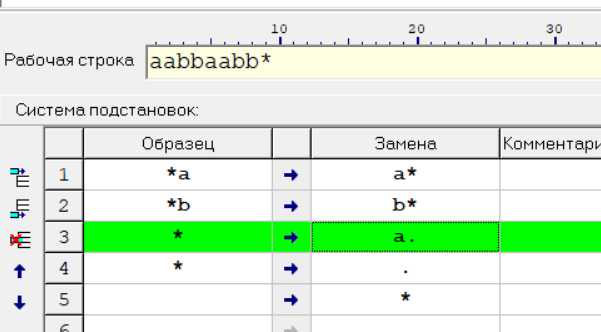


1. \*а заменяем на а\*, \*b заменяем на b\* (благодаря этому перемещаем \* в конец слова);



1. \* заменяем на а, алгоритм завершается;



1. Если слово пустое, то \* удаляется, алгоритм завершается.