

ЗАДАНИЕ
на лабораторную работу № 4
по дисциплине «Теория вычислительных процессов и структур»

**Проектирование нормальных алгоритмов Маркова,
реализующих простые текстовые операции.**

Время: 2 часа (90 минут).

Учебные цели:

1. Выработать у студентов практические умения и навыки в построении НАМ, в том числе с помощью симуляторов.

2. Формировать способности: применять современный математический аппарат, связанный с проектированием, разработкой, реализацией и оценкой качества программных продуктов и программных комплексов в различных областях человеческой деятельности (ОПК-2); использовать основные концептуальные положения функционального, логического, объектно-ориентированного и визуального направлений программирования, методы, способы и средства разработки программ в рамках этих направлений (ПК-6)

Программный симулятор доступен по ссылке:

<https://kpolyakov.spb.ru/prog/nma.htm>

Пароль к архиву – kpolyakov.spb.ru

Марковской подстановкой называется операция над словами, задаваемыми с помощью упорядоченной пары слов (P, Q) , состоящая в следующем. В заданном слове R находят первое вхождение слова P (если оно есть) и, не изменяя остальных частей слова R , заменяют в нем это вхождение словом Q . Полученное слово называется результатом применения марковской подстановки (P, Q) к слову R . Если же первого вхождения P в слово R (и, следовательно, вообще нет ни одного вхождения P в R), то считается, что марковская подстановка (P, Q) не применима к слову R . Запись $P \rightarrow Q$ называется формулой подстановки (P, Q) . P называется левой частью, Q – правой частью в формуле подстановки. Некоторые подстановки называются заключительными. Для обозначения таких подстановок будем использовать запись $P \rightarrow (\cdot)Q$, называя ее формулой заключительной подстановки.

Упорядоченный конечный список формул подстановок в алфавите A

$$\left\{ \begin{array}{l} P_1 \rightarrow (\cdot)Q_1 \\ | \\ P_2 \rightarrow (\cdot)Q_2 \\ | \\ : \\ | \\ P_n \rightarrow (\cdot)Q_n \end{array} \right.$$

называется схемой нормального алгоритма в алфавите A . Запись точки в скобках означает, что она может стоять на этом месте, а может отсутствовать.

Пример1. Построить нормальный алгоритм Маркова, заменив в алфавите $A = \{a, b, c\}$ все буквы a на c .

Решение:

Используем символ α для расширения алфавита A . $B = \{\alpha\} \cup A$. Схема Z нормального алгоритма будет иметь следующий вид:

$$Z: \begin{cases} \alpha a \longrightarrow c\alpha \\ \alpha b \longrightarrow b\alpha \\ \alpha c \longrightarrow c\alpha \\ \alpha \longrightarrow \cdot \Lambda \\ \Lambda \longrightarrow \alpha \end{cases}$$

Например, $aacbab \Rightarrow \alpha aacbab \Rightarrow c\alpha acbab \Rightarrow cc\alpha cbab \Rightarrow ccc\alpha bab \Rightarrow cccb\alpha ab \Rightarrow cccbc\alpha b \Rightarrow cccbc\alpha b \Rightarrow cccbc\alpha b \Rightarrow cccbc\alpha b$.

Этот алгоритм может быть реализован так же следующей схемой:

$$Z1: \begin{cases} a \rightarrow c \\ \Lambda \rightarrow \cdot \Lambda \end{cases}$$

Пример2. Дано слово в алфавите $A = \{a, b, c\}$. Построить алгоритм Маркова, присоединяющий слово Q к данному слову.

Решение:

$$\begin{cases} \varepsilon a \rightarrow a\varepsilon \\ \varepsilon b \rightarrow b\varepsilon \\ \varepsilon c \rightarrow c\varepsilon \\ \varepsilon \rightarrow \cdot Q \\ \Lambda \rightarrow \varepsilon \end{cases}$$

Вариант №1

Задача №1.

Построить нормальный алгоритм Маркова, который бы в слове из алфавита $A = \{a, b, c, d\}$ заменял первое вхождение подслоа bb на ddd и удалял все вхождения символа c .

Задача №2.

$A = \{a, b\}$. Преобразовать слово P так, чтобы в его начале оказались все символы a , а в конце – все символы b .

Задача №3.

Построить нормальный алгоритм Маркова, который бы в слове из алфавита $A = \{a, b, c, d, e, f\}$ все вхождения последовательности abc заменял на символ f и удалял первое вхождение пары cf .

Задача №4.

Построить нормальный алгоритм Маркова, который бы в слове из алфавита $A = \{a, b, c, d, e, f\}$ удалял все вхождения последовательности bc и удваивал гласные буквы.

Задача №5.

Построить нормальный алгоритм Маркова, который бы в слове из алфавита $A = \{a, b, c, d, e, f\}$ все символы a заменял на f , а все f – на af .

Задача №6.

$A = \{a, b, c\}$. Заменить слово P на пустое слово, т.е. удалить из P все символы.

Задача №7.

$A = \{a, b, c\}$. Определить, входит ли символ a в слово P . Ответ (выходное слово): слово a , если входит, или пустое слово, если не входит.

Задача №8.

$A = \{a, b\}$. Перевернуть слово P (например: $abb \rightarrow bba$).

Вариант №2**Задача №1.**

$A = \{a, b\}$. Преобразовать слово P так, чтобы в начале оказались все символы a , а в конце – все символы b .

Задача №2.

$A = \{a, b, c\}$. Приписать слово bac слева, к слову, P

Задача №3.

Построить нормальный алгоритм Маркова, который бы в слове из алфавита $A = \{a, b, c, d, e, f\}$ все вхождения последовательности cde заменял на символ a и удваивал согласные буквы.

Задача №4.

$A = \{a, b, c\}$. Заменить любое входное слово на слово a .

Задача №5.

Выписать НАМ, не меняющий входное слово (при любом алфавите A).

Задача №6.

$A = \{a, b\}$. Удвоить слово P , т.е. приписать к P (слева или справа) его копию.

Задача №7.

Построить нормальный алгоритм Маркова, который бы в слове из алфавита $A = \{a, b, c, d, e, f\}$ все символы e заменял на d , а все d – на de .

Задача №8.

Построить нормальный алгоритм Маркова, который упорядочивает любое слово в алфавите $A = \{a, b, c, d\}$.