

Нормальные алгоритмы Маркова на ПЭВМ

Нормальный алгоритм Маркова задается алфавитом $A = \{a_1, a_2, \dots, a_n\}$, нормальной схемой подстановок и порядком их применения. Схема подстановок представляет собой набор формул вида $P_k \rightarrow P_r$, согласно которым подслово P_k в исходном слове P заменяется словом P_r . Если подстановка возможна, то ее осуществляют и все начинают сначала. Обычно студенты “вручную” реализуют нормальную схему подстановок, в результате чего исходное слово P превращается в слово Q . В качестве творческого задания преподаватель может предложить студентам написать программу, реализующую нормальную схему подстановок. Листинг программы представлен ниже (<http://maier-rv.glazov.net>).

```
uses crt, graph;                                { Borland Pascal 7.0 }
const Chislo_podstan=10; label m1;
var i,j,k,m,p,s,flag: integer; x1,x2,x4,x5,x6,q: string;
    x,y: array[1..100] of string; slovo, slovo1: string;
Procedure Podstanovka(j:integer); label m;
begin flag:=0; i:=0;
Repeat inc(i);
  If copy(slovo,i,length(x[j]))=x[j] then begin flag:=1;
    slovo1:=copy(slovo,1,i-1)+y[j]+copy(slovo,
    i+length(x[j]),length(slovo)-i-length(x[j])+1);
    slovo:=slovo1; If (x[j]='')or(y[j]='') then flag:=0;
    goto m; end;
until i>length(slovo); m:
If flag=1 then writeln(k,' ',slovo,' | подстановка ',j);
end;
BEGIN clrscr; slovo:='B_AB_BA_AA_BA_BB_A_BA';
writeln(slovo); x[1]:='BA'; y[1]:='AB'; x[2]:='B_';
y[2]:='_B'; x[3]:='_A'; y[3]:='A_';
m1: k:=k+1; delay(5000);
For p:=1 to Chislo_podstan do begin
  Podstanovka(p); If flag=1 then goto m1; end; Readkey;
END.
```

Представленная программа позволяет решить следующие задачи:

Задача 1. Имеется слово 'B_AB_VA_AA_VA_BB_A_VA'. Создайте нормальный алгоритм Маркова, который символы 'A' переносит влево, символы 'B' — вправо, а пробелы оставляет посередине. Промоделируйте исполнение алгоритма на компьютере. Решение: 1) 'BA' => 'AB'; 2) 'B_' => '_B'; 3) '_A' => 'A_'.

Задача 2. Имеется слово 'ABCDVEACBDEABCE'. Создайте нормальный алгоритм Маркова, который осуществляет побуквенное кодирование этого слова. Промоделируйте его исполнение на компьютере. Решение: 1) 'A' => '00-'; 2) 'B' => '01-'; 3) 'C' => '10-'; 4) 'D' => '11-'; 5) 'E' => '111-'.

Задача 3. Убедитесь в возможности построения бесконечного алгоритма Маркова. Для этого промоделируйте на компьютере следующую схему подстановок: 1) 'CA'=>'CC'; 2) 'CCB'=>'BA'; 3) 'BA'=>'ACB'. Исходное слово: 'ACBCBAC'.

Задача 4. Создайте нормальный алгоритм Маркова, осуществляющий сложение двух чисел от 1 до 9. Промоделируйте его работу на компьютере.

Решение: 1) '1one' => '|'; 2) '2two' => '||'; 3) '3three' => '|||'; 4) '4four' => '||||'; 5) '5five' => '|||||'; 6) '6six' => '||||||'; 7) '7seven' => '|||||||'; 8) '8eight' => '||||||||'; 9) '9nine' => '|||||||||'; 10) '|+' => '||'; 11) '|||||||||' => '10'; 12) '0|||||||||' => '9'; 13) '0||||||||' => '8'; 14) '0||||||||' => '7'; 15) '0|||||||' => '6'; 16) '0||||||' => '5'; 17) '0|||||' => '4'; 18) '0|||' => '3'; 19) '0||' => '2'; 20) '0|' => '1'; 21) '|||||||||' => '9'; 22) '||||||||' => '8'; 23) '|||||||' => '7'; 24) '||||||' => '6'; 25) '|||||' => '5'; 26) '||||' => '4'; 27) '|||' => '3'; 28) '||' => '2'; 29) '|' => '1'. Исходное слово записывается в виде: '8eight+5five' или '9nine+7seven'.

Использование рассмотренной методики при изучении курса “Теоретические основы информатики”, в курсовых и дипломных проектах способствует более глубокому пониманию сущности алгоритмов Маркова, повышению интереса студентов к информатике и программированию.

Приложение

1. Могилев А.В. Информатика: Учеб. пособие для студ. пед. вузов [Текст] / Могилев А.В., Пак Н.И., Хеннер Е.К. — М.: Издательский центр “Академия”, 2003. — 816 с.