

Akademia Nauk Stosowanych - Teoretyczne i technologiczne podstawy multimediiów - laboratorium			
Temat: Kodowanie arytmetyczne.			Symbol: TiTPM
Nazwisko i imię: Fyda Kamil		Ocena sprawozdania	Zaliczenie:
Data wykonania ćwiczenia: 8.11.2022r.		Grupa: L1	

1. Opis teoretyczny:

Kodowanie arytmetyczne – metoda kodowania źródłowego dyskretnych źródeł sygnałów, stosowana jako jeden z systemów w bezstratnej kompresji danych. Została wynaleziona przez Petera Eliasa około 1960 roku. Od roku 2014 było zastępowane kodowaniem Asymmetric Numeral Systems, które pozwala na szybsze implementacje przy podobnym stopniu kompresji.

Ideą tego kodu jest przedstawienie ciągu wiadomości jako podprzedziału przedziału jednostkowego (0,1) wyznaczonego rekursywnie na podstawie prawdopodobieństw wystąpienia tychże wiadomości generowanych przez źródło. Ciąg kodowy reprezentujący kodowane wiadomości jest binarnym zapisem wartości z wyznaczonego w ten sposób przedziału.

Można udowodnić, że przy wyborze odpowiednio długiego ciągu wiadomości do zakodowania, średnia liczba symboli na wiadomość jest mniejsza od $H(X)+2$ gdzie $H(X)$ jest entropią źródła, lecz co najmniej równa samej entropii.

2. Założenia:

- Komunikat składa się z e znaków alfabetu $A = \{a_1, a_2, a_3, a_4, \dots, a_n\}$
- Znane jest prawdopodobieństwo użycia każdego ze znaków $P = \{p_1, p_2, p_3, p_4, \dots, p_n\}$

3. Przykład graficzny Algorytmu arytmetycznego:

• ABRAADABRA

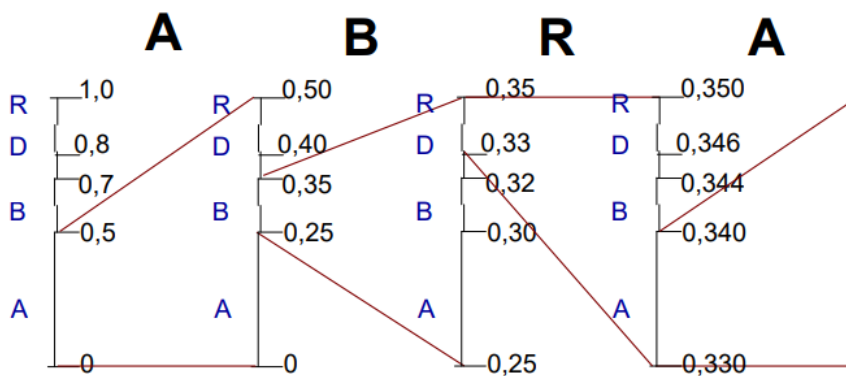
Wszystkich znaków 10

A – 5 $\rightarrow [0; 0,5)$ bo 5/10

B – 2 $\rightarrow [0,5; 0,7)$ bo 2/10

D – 1 $\rightarrow [0,7; 0,8)$ bo 1/10

R – 2 $\rightarrow [0,8; 1)$ bo 2/10



4. Kod programu:

```
kodowanie.cpp
1  #include <iostream>
2  #include <string>
3  #include <algorithm>
4  #include <map>
5  #include <vector>
6
7  using namespace std;
8
9  vector<float> mnoznik;
10
11
12 void wypisanie(vector<char> znaki, map<char, float> mymap, string text, int licznik, float mapFirtstValue)
13 {
14     cout<<endl;
15     if(licznik<text.length())
16     {
17         if(licznik == 0)
18         {
19             for(int i=0; i<znaki.size(); i++)
20             {
21                 cout<<mymap.find(znaki[i])>>first <<" -> " <<mymap.find(znaki[i])>>second<<endl;
22             }
23         }
24         float newPartition;
25         float tempMap = mapFirtstValue;
26         for(int i=0; i<znaki.size(); i++)
27         {
28             cout<<"[" << tempMap <<" "; tempMap += (mymap.find(znaki[i])>>second)<<"]" <<endl;
29             if(text[licznik] == mymap.find(znaki[i])>>first)
30             {
31                 newPartition = mymap.find(znaki[i])>>second;
32                 mapFirtstValue = tempMap;
33             }
34             tempMap += mymap.find(znaki[i])>>second;
35         }
36     }
37 }
38
39
40
41 int main()
42 {
43     string text, text_temp;
44     cout<<"Podaj slowo do zakodowania:"<<endl;
45     cin>>text;text_temp = text;
46     sort(text_temp.begin(), text_temp.end());
47     cout<<endl<<"Uzyte znaki w tym slowie: "<<endl<<text_temp<<endl;
48
49     map<char, float> mymap;
50     vector<char> znaki;
51
52     char znak = text_temp[0];
53     int counter = 0;
54     for(int i=0; i<text_temp.length(); i++)
55     {
56         if(i == text_temp.length()-1 && znak != text_temp[i])
57         {
58             mymap.insert(pair<char,float>(znak, float(counter)/float(text.length())));
59             znaki.push_back(znak);
60             mnoznik.push_back(float(counter)/float(text.length()));
61             mymap.insert(pair<char,float>(text_temp[i], float(1)/float(text.length())));
62             znaki.push_back(text_temp[i]);
63             mnoznik.push_back(float(counter)/float(text.length()));
64             break;
65         }
66         if(znak != text_temp[i] || i == text_temp.length()-1)
67         {
68             if(i == text_temp.length()-1) counter++;
69             mymap.insert(pair<char,float>(znak, float(counter)/float(text.length())));
70             znaki.push_back(znak);
71             mnoznik.push_back(float(counter)/float(text.length()));
72             counter = 0;
73             znak =text_temp[i];
74         }
75         counter++;
76     }
77
78     wypisanie(znaki, mymap, text, 0, 0);
79
80     return 0;
81 }
82 }
```

5. Wynik działania programu:

```
C:\Users\Asus\Desktop\kodowanie.exe
Podaj słowo do zakodowania:
ABRAADABRA

Użyte znaki w tym słowie:
AAAAABBD RR

A -> 0.5
B -> 0.2
D -> 0.1
R -> 0.2
[0;0.5)
[0.5;0.7)
[0.7;0.8)
[0.8;1)

-----
Process exited after 121.4 seconds with return value 0
Press any key to continue . . .

C:\Users\Asus\Desktop\kodowanie.exe
Podaj słowo do zakodowania:
cojatutajrobie

Użyte znaki w tym słowie:
aabceijjoorttu

a -> 0.142857
b -> 0.0714286
c -> 0.0714286
e -> 0.0714286
i -> 0.0714286
j -> 0.142857
o -> 0.142857
r -> 0.0714286
t -> 0.142857
u -> 0.0714286
[0;0.142857)
[0.142857;0.214286)
[0.214286;0.285714)
[0.285714;0.357143)
[0.357143;0.428571)
[0.428571;0.571429)
[0.571429;0.714286)
[0.714286;0.785714)
[0.785714;0.928571)
[0.928571;1)

-----
Process exited after 6.399 seconds with return value 0
Press any key to continue . . .
```

Wnioski:

Jak widzimy powyżej program realizuje poprawnie założenia zadania. Można zatem powiedzieć, iż implementacja Algorytmu kodowania arytmetycznego działa w sposób jak najbardziej prawidłowy.