Autor: Klaudia Miękina

Kodowanie i transmisja danych – ćwiczenia nr 6

Na ćwiczeniach laboratoryjnych omówiono algorytm Huffmana.

Algorytm Huffmana jest jedną z bardziej skutecznych metod bezstratnej kompresji danych, polegającej na zastępowaniu symboli występujących w strumieniu wejściowym specjalnymi sekwencjami bitów stanowiących tzw. słowa kodowe. Słowa kodowe dobiera się w taki sposób, by najkrótsze z nich przyporządkować symbolom najczęściej występującym w strumieniu wejściowym, a najdłuższe symbolom o najniższych częstotliwościach wystąpień. Ponadto żadna sekwencja bitów nie może być przedłużeniem innej, co zapewnia jednoznaczność pomiędzy reprezentacją naturalną a zbiorem danych przedstawionym w postaci zakodowanej.[1]

Poniżej przedstawiono kilka tekstów (zdań) różnej długości oraz przeprowadzono następujące działania:

- utworzono kod zmiennorozmiarowy wykorzystując kodowanie Huffmana;
- określono bezwzględny i względny zysk binarny w porównaniu do zastosowania kodu stałorozmiarowego;
- sprawdzono, czy utworzony kod jest jednoznacznie dekodowalny.

Dodatkowo:

- pokazano drzewo powstałe po zakończeniu kodowania Huffmana;
- jako stałorozmiarowy kod rozważono dwa przypadki:
  - 8-bitowy kod ASCII,
  - obliczony minimalny rozmiar kodów binarnych stosownie do liczby różnych symboli w zdaniu (np. dla 5 symboli będzie to rozmiar 3-bitowy).

1. Dla zdania „skocik piesek" wartości unikalne posortowane rosnąco:

o: 1
c: 1
(spacja) : 1
p: 1
s: 2
i: 2
e: 2
k: 3
Liczba elementów: 8

Drzewo powstałe po zakończeniu kodowani Huffmana:

o: 1 c: 1 (spacja): 1 p: 1 s: 2 i: 2 e: 2 k: 3

\1 / 0 \1 / 0 \1 /0 \1 /0

?:2 ?:2 ?:4 ?:5

\1 / 0 \1 /0

?:4 ?:9

\1 /0

?:13

[1] http://ics.p.lodz.pl/~dpuchala/PodstInfII/projektII_opis.pdf

Słowa kodowe:
o: 111
c: 110
' ': 101
p: 100
s: 011
i: 010
e: 001
k: 000
R1 = 1 * 3 + 1 * 3 + 1 * 3 + 2 * 3 + 2 * 3 + 3 * 3 = 3 + 3 + 3 + 6 + 6 + 9 = 30
R0 = 4 * 13 = 52

Bezwzględny zysk binarny **k = R1 / R0 = 30 / 52 = 0,57**
Względny zysk binarny **K = 52 - 30 / 52 = 0,43**
Sprawdzono słowa kodowe pod względem jednoznacznej dekodowalności za pomocą skryptu. Kod jest jednoznacznie dekodowalny.
Przypadek: 8-bitowy ASCII: 13 * 8 = 104
Przypadek: minimalny rozmiar kodów binarnych stosownie do liczby różnych symboli w zdaniu: 3 bity

2. Dla zdania „włoski lekkoatleta specjalizujący się w biegu na 400 metrów" wartości unikalne posortowane rosnąco:

ł: 1
p: 1
z: 1
ą: 1
y: 1
ę: 1
b: 1
g: 1
n: 1
4: 1
m: 1
r: 1
ó: 1
o: 2
c: 2
j: 2
u: 2
0: 2
w: 3
s: 3
k: 3
l: 3
t: 3
i: 4
a: 4
e: 5
(spacja) : 8
Liczba elementów: 27
Drzewo powstałe po zakończeniu kodowani Huffmana:
ł: 1 p: 1 z: 1 ą: 1 y: 1 ę: 1 b: 1 g: 1 n: 1 4: 1 m: 1 r: 1 ó: 1 o: 2 c: 2 j: 2 u: 2 0: 2 w: 3 s: 3 k: 3 l: 3 t: 3 i: 4 a: 4 e: 5 (spacja) : 8

Słowa kodowe:
ł 11111
p 11110
z 11101
ą 11100
y 11011
ę 11010
b 11001
g 11000
n 10111
4 10110
m 10101
r 10100
ó 10011
o 10010 2
c 10001
j 10000

u 01111
0 01110
w 01101 3
s 01100
k 01011
l 01010
t 01001
i 01000 4
a 00011
e 00010 5
spacja 00000

R1 = 1 * 5 + 1 * 5 + 1 * 5 + 1 * 5 + 1 * 5 + 1 * 5 + 1 * 5 + 1 * 5 + 1 * 5 + 1 * 5 + 1 * 5 + 1 * 5 + 1 * 5 + 1 * 5 + 2 * 5 + 2 * 5 + 2 * 5 + 2 * 5 + 2 * 5 + 3 * 5 + 3 * 5 + 3 * 5 + 3 * 5 + 3 * 5 + 4 * 5 + 4 * 5 + 5 * 5 = 255
R0 = 6 * 59 = 354

Bezwzględny zysk binarny **k = R1 / R0 = 255 / 354 = 0,72**
Względny zysk binarny **K = 354 - 255 / 354 = 0,28**
Sprawdzono słowa kodowe pod względem jednoznacznej dekodowalności za pomocą skryptu. Kod jest jednoznacznie dekodowalny.
Przypadek: 8-bitowy ASCII: 59 * 8 = 472
Przypadek: minimalny rozmiar kodów binarnych stosownie do liczby różnych symboli w zdaniu: 6 bitów