Ćwiczenie 4

Metody kompresji informacji w sieci

Dynamiczna metoda Huffmana

Aby wykonać kodowanie Huffmana należy znać prawdopodobieństwa (częstość występowania) liter. Jeżeli nie zna się statystyk, a dane są dostarczane na bieżąco, należy kodować *k+1* symbol na podstawie statystyk *k* symboli. Dla kodu Huffmana tworzy się na bieżąco jego drzewo o następujących właściwościach:

- każdy liść odpowiada symbolowi i zawiera wagę-ilość dotychczasowych wystąpień,
- wierzchołki wewnętrzne mają wagę będącą sumą wag liści z poddrzew,
- każdy wierzchołek drzewa ma unikalny numer x_i. Numery te tworzą porządek zgodny z wagami wierzchołków (większa waga to większy numer wierzchołka). Dodatkowo rodzeństwo ma zawsze dwa kolejne numery.

Na początku drzewo zawiera jeden wierzchołek o wadze 0 i etykiecie NYT oznaczającej, że symbol nie był jeszcze przesyłany.

Opis algorytmu

1. Pierwsze wystąpienie symbolu *a*
   a) Wyślij kod NYT i kod stałej długości dla nowego symbolu.
   b) Stary NYT podziel na dwa wierzchołki potomne - nowy NYT i liść *a*, nadaj *a* wagę 1. Nadaj im odpowiednie numery.
   c) Zmodyfikuj drzewo dodając 1 do wierzchołków wewnętrznych na ścieżce od *a* do korzenia *i* przebudowując drzewo tak aby było zgodne z warunkami powyżej
2. Kolejne wystąpienie symbolu *a*
   a) Znajdź liść *a* i wyślij odpowiadający mu kod.
   b) Zwiększ wagę *a* o 1.
   c) Zmodyfikuj drzewo dodając 1 do wierzchołków wewnętrznych na ścieżce od *a* do korzenia *i* przebudowując drzewo tak aby było zgodne z poprzednimi warunkami

Modyfikacja drzewa

- Zbiór wierzchołków o tej samej wadze nazywa się blokiem.
- Jeśli pierwszy wierzchołek od dołu nie ma największego numeru w swoim bloku to zamieniamy go z tym o największym numerze odpowiednio przebudowując drzewo z zachowaniem własności. Następnie aktualizuje się wagę i patrzy dalej rekurencyjnie.
- Koniec nastąpi gdy dojdziemy do korzenia.

Istnieją dwa algorytmy pozwalające poprawić drzewo Huffmana:

1. **algorytm Fallera-Gallera-Knutha** (pomysłodawcami byli Newton Faller i Robert Galler, metodę ulepszył Donald Knuth),
2. **algorytm Vittera** (dalsze ulepszenia metody FGK opracowane przez Jeffreya Vittera).

Projekt Absolwent na miarę czasu współfinansowany przez Unię Europejską w ramach Europejskiego Funduszu Społecznego;
Nr umowy UDA-POKL.04.01.01-00-421/10-01

U podstaw **algorytmu FGK** leżą następujące założenie co do formy drzewa:

- każdy węzeł drzewa oprócz liści ma zawsze **dwóch potomków**;
- z każdym węzłem związany jest licznik: w liściach przechowuje liczbę wystąpień danego symbolu (lub wartość *proporcjonalną*), w pozostałych węzłach sumę liczników dzieci;
- przy przejściu drzewa wszerz od prawej do lewej i odczycie liczników powiązanych z każdym węzłem uzyskuje się ciąg liczb nierosnących.

A**lgorytm Vittera** zaostrzone zostało ostatnie założenie:

- również otrzymuje się ciąg liczb nierosnących, lecz w obrębie podciągów o tych samych wartościach na początku znajdują się te pochodzące z węzłów wewnętrznych, a na końcu z liści.

Gdy licznik w jakimś liściu zwiększy się, algorytmy modyfikują (przemieszczając niektóre węzły) jedynie niewielki fragment drzewa, zachowując wyżej wymienione własności. **Algorytm Vittera** jest nieco bardziej złożony, jednak daje lepsze wyniki, tj. krótsze kody, niż **algorytm FKG**.

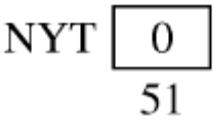
Przykład

Występują cztery litery z 26. (W drzewie maksymalnie 51 wierzchołków.)

a d r v

00001 00100 10010 10110

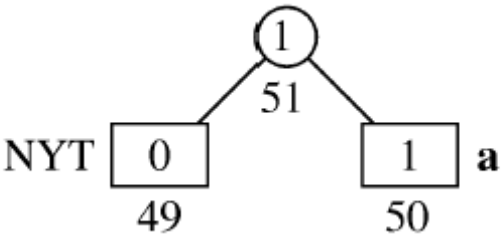
Drzewo początkowo wygląda tak:



Pojawia się litera a.

Zostaje wysłany kod NYT (ε) i stały kod a: 00001.

Następuje modyfikacja drzewa:

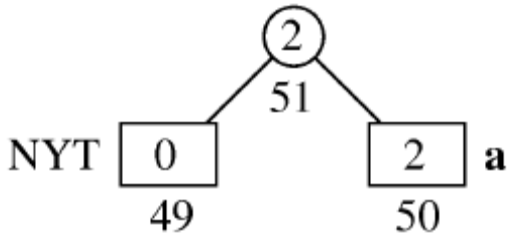


Pojawia się druga litera a.

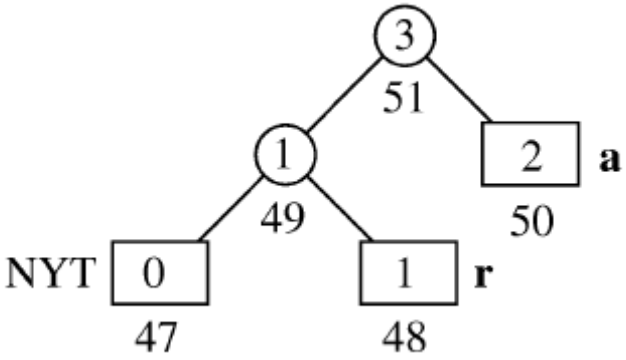
Zostaje wysłany kod a: 1 (z drzewa lewe krawędzie to 0 a prawe to 1).

Następuje modyfikacja drzewa:

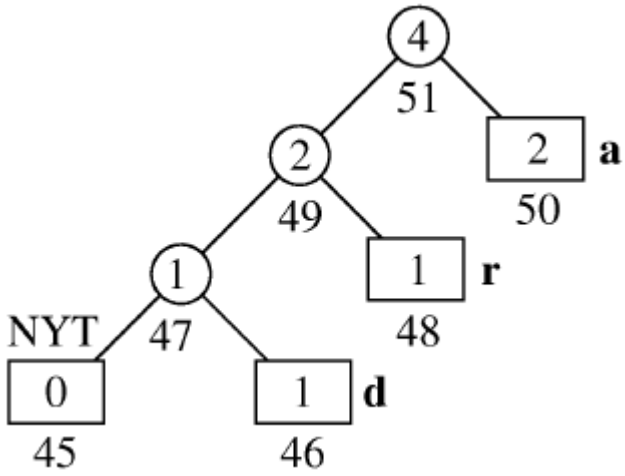
Projekt Absolwent na miarę czasu współfinansowany przez Unię Europejską w ramach Europejskiego Funduszu Społecznego;
Nr umowy UDA-POKL.04.01.01-00-421/10-01



Pojawia się litera r .
Zostaje wysłany NYT (0) i stały kod r : 010010.
Następuje modyfikacja drzewa:

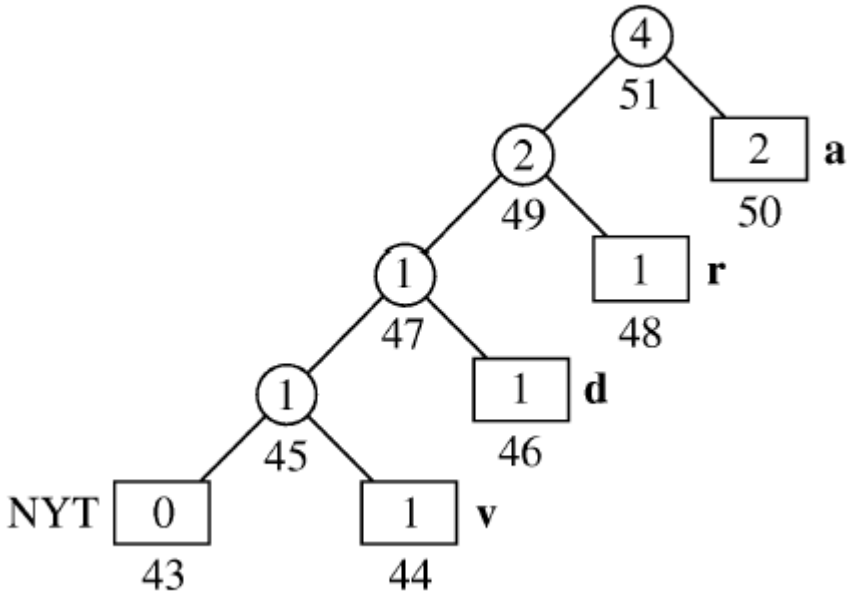


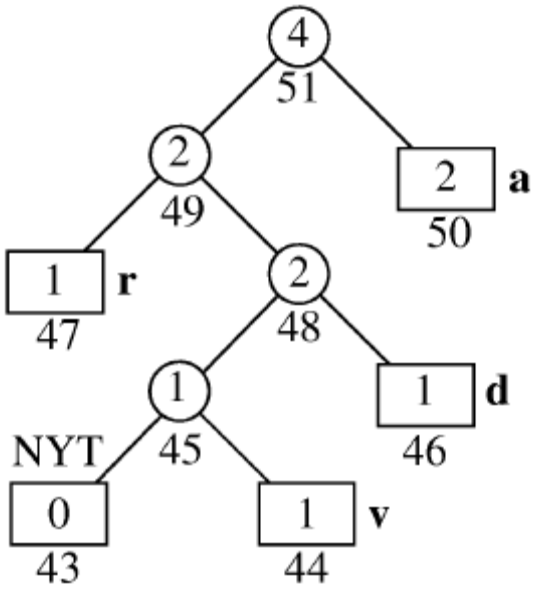
Pojawia się litera d.
Zostaje wysłany NYT (00) i stały kod d: 0000100.
Następuje modyfikacja drzewa:



Projekt Absolwent na miarę czasu współfinansowany przez Unię Europejską w ramach Europejskiego Funduszu Społecznego;
Nr umowy UDA-POKL.04.01.01-00-421/10-01

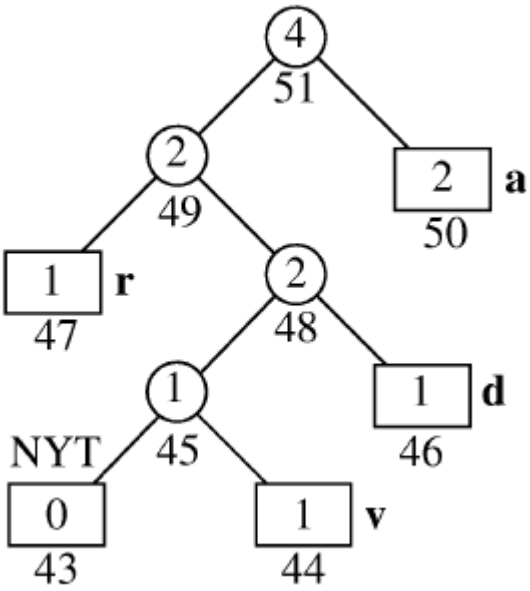
Pojawia się litera v.
Zostaje wysłany NYT (000) i stały kod v: 0010110.
Następuje modyfikacja drzewa:

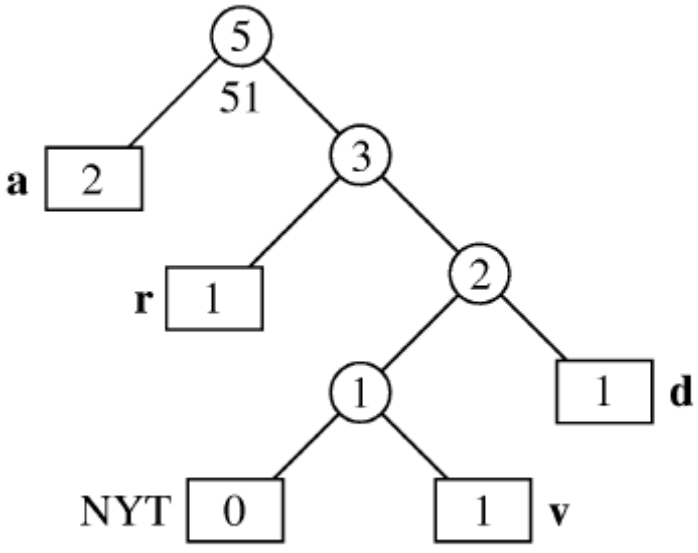




Następuje modyfikacja drzewa

Projekt Absolwent na miarę czasu współfinansowany przez Unię Europejską w ramach Europejskiego Funduszu Społecznego;
Nr umowy UDA-POKL.04.01.01-00-421/10-01





Stworzyć algorytm który dla dowolnego tekstu podawanego przez prowadzącego skompresuje dane dynamicznym kodowaniem Huffmana.

Projekt Absolwent na miarę czasu współfinansowany przez Unię Europejską w ramach Europejskiego Funduszu Społecznego;
Nr umowy UDA-POKL.04.01.01-00-421/10-01

Następnie wyznaczyć wartość entropii wg zależności:

$$H(\ell) = \sum_{i=1}^{n} p_i * \log_2\left(\frac{1}{p_i}\right)$$

gdzie p_i jest prawdopodobieństwem wystąpienia i-tego znaku.

Mając określone słowa kodowe wyznaczyć średnią długość słowa kodowego według zależności:

$$\bar{L} = \sum_{i=1}^{n} p_i * m_i$$

gdzie m_i jest ilością bitów w słowie kodowym

Sprawdzić jaka jest zależność pomiędzy entropią a średnią długością słowa kodowego. Wyjaśnić otrzymane wyniki.

Projekt Absolwent na miarę czasu współfinansowany przez Unię Europejską w ramach Europejskiego Funduszu Społecznego;
Nr umowy UDA-POKL.04.01.01-00-421/10-01