

DSP - Cyfrowe przetwarzanie sygnałów

Laboratorium 5–7 – Dyskretna transformata Fouriera

Jakub Moroń, Paweł Hottowy

Utworzono: Kwiecień 2022

Ostatnia zmiana: 3 kwietnia 2023

Zdania oznaczone (*) zawierają elementy do zamieszczenia w raporcie.

Spis treści

1 Podstawy DFT	3
1.1 Wprowadzenie do DFT(*)	3
1.2 Przesunięcie fazowe	3
2 Widmo sygnału prostokątnego	4
2.1 Widmo impulsu prostokątnego(*)	4
2.2 Widmo sygnału prostokątnego	4
2.3 Współczynnik wypełnienia(*)	5
3 Wyciek DFT	5
4 Widmo instrumentów muzycznych	5
5 Okna DFT	5
6 Rozdzielczość transformaty DFT	6
6.1 Rozdzielczość a wyciek(*)	6
6.2 Rozdzielczość a szum	6
7 Transformata odwrotna – filtracja w dziedzinie częstotliwości	7
7.1 Filtracja sygnału prostokątnego(*)	7

7.2	Filtracja muzyki	7
7.3	Filtracja przydźwięku sieciowego	7
8	Transformata DFT obrazu	7
8.1	Podstawy DFT obrazu – pionowe i poziome linie	7
8.2	Podstawy DFT obrazu – animacja skośnych linii	8
8.3	Podstawy DFT obrazu – obrazy bardziej złożone(*)	8
8.4	Podstawy DFT obrazu – widmo okręgu	8
8.5	DFT realnych zdjęć	8
9	Filtracja zdjęć w dziedzinie częstotliwości	8
9.1	Filtracja dolnoprzepustowa zdjęć(*)	8
9.2	Filtracja górnoprzepustowa zdjęć	9
9.3	Filtracja pasmowa zdjęć	9
9.4	Operatory Sobel’a	9
9.5	Prosty algorytm wyostrzający(*)	9
10	Kompresja dźwięku i obrazu	10
10.1	Kompresja „losowa”	10
10.2	MDCT – spektrogram(*)	10
10.3	Kompresja mowy	10
10.4	Kompresja obrazu – JPEG(*)	11
10.5	Kompresja bezstratna – kodowanie Huffman’a(*)	11

1 Podstawy DFT

1.1 Wprowadzenie do DFT(*)

Skrypt Zad1_1_podstawy_DFT.

Wykonaj skrypt. Następnie:

- Przedyskutuj zawartość wykresów, w szczególności:
 - Szczególną uwagę zwróć na wykres fazy - czy przedstawia to co powinien? Dlaczego?
 - Czy wysokość prążka dla sinusa zgadza się z jego amplitudą Amp?
- Usuń komentarz z linii 41. Przedyskutuj:
 - Czy teraz wykres fazy zgadza się z teorią?
 - Jaką symetrię ma widmo amplitudowe i fazowe? Dla jakich sygnałów to zachodzi?
- Wyraź wykresy 2) i 3) w funkcji częstotliwości, a nie numeru prążka. W tym celu uzupełnij odpowiednio wyrażenie na Fbase (linia 29.)
- Unormuj amplitudę widma - uzupełnij odpowiednio linię 30. tak, aby zmienna amplitudeNorm zawierała unormowane widmo.
- Zanotuj fazę dla prążka sygnału. Zmień początkową fazę sygnału (zmienna Phase) na 0.5, 1.0 - jaka faza wychodzi dla prążka sygnałowego i jak się ma do wartości Phase? Dlaczego?
- Dodaj składową stałą ustawiając $DC = 2$. W którym prążku zawarta jest składowa stała? Czy jej wartość się zgadza? Jeśli nie, to należy poprawić normowanie.
- Zmień częstotliwość sinusa (zmienna Fsig) na 2500 Hz. Czy rozumiesz co się stało?

Raport

1. Przedstaw wykresy dla przypadku sinusa ze składową stałą. Wykres przedstaw w skali częstotliwości wyrażonej w [Hz] oraz prawidłowo unormowany.
2. Przedstaw:
 - a) wzór (wzory) na normowanie amplitudy,
 - b) wzór na obliczenie częstotliwości bazowej oraz wartość częstotliwości bazowej w ćwiczeniu.
3. Podaj wartości faz dla prążków sinusa i kosinusa (sinusa o fazie 0.5π) i zinterpretuj teoretycznie.

1.2 Przesunięcie fazowe

Skrypt Zad1_2_przesuniecie_fazowe.

- Przeanalizuj i przedyskutuj wykresy - jak zmieniła się faza pików sygnału dla $N_{skip} = 0$ oraz $N_{skip} = 4$

- Na podstawie wykładu uzasadnij przesunięcie fazowe. Oblicz fazę Phase sygnału jaką należy użyć dla przypadku Nskip = 4 aby uzyskać dokładnie taki sam wynik jak dla Phase = 0 oraz Nskip = 0.

2 Widmo sygnału prostokątnego

2.1 Widmo impulsu prostokątnego(*)

Skrypt Zad2_1_widmo_box.

Literatura: Richard G. Lyons - Understanding Digital Signal Processing (2nd Edition)
- Chapter Three. The Discrete Fourier Transform: The DFT of rectangular functions.

Analityczna postać m -tego prążka jądra Dirichlet'a dla sygnału o szerokości K , położonego n_0 od środka ciągu próbek o długości N dana jest jako

$$X(m) = \frac{\sin\left(\pi \frac{mK}{N}\right)}{\sin\left(\pi \frac{m}{N}\right)} \cdot e^{j\left(2\pi \frac{m}{N}\right)\left(-n_0 - \frac{K-1}{2}\right)} \quad (1)$$

- Przedyskutuj wykresy. Porównaj zgodność wyniku DFT z funkcjami sinc oraz jądrem Dirichlet'a (szczególnie zwróć uwagę na główny pik)
- Wiedząc że pierwsze minimum funkcji sinc/jądra Dirichlet'a położone jest w pozycji N/K , sprawdź zgodność wyniku z teorią.
- Sprawdź czy widmo amplitudowe zależy od położenia prostokąta w ciągu próbek, zmieniając n_0 (przy stałym K)
Zakomentuj linie 39. oraz 45. aby wykonać skrypt w całości bez zatrzymywania.
- Sprawdź jaka jest zależność szerokości głównego piku widma od szerokości prostokąta K , zmieniając K .
- Ustaw $K=1$ (impuls) - przedyskutuj widmo.
- Ustaw $K=N$ (użyj $n_0 = -(K-1)/2$) - przedyskutuj widmo.

Raport

1. Przedstaw wykres dla $K=8$, $n_0=-5$. Omów zgodność wyniku DFT z oboma funkcjami teoretycznymi.
2. Podaj odczytane z wykresu położenie pierwszego minimum powyżej piku głównego i porównaj z wartością teoretyczną.
3. Przedstaw wniosek o zależności widma amplitudowego od n_0 .
4. Przedstaw widmo dla $K=N$ i $n_0 = -(K-1)/2$. Przedstaw (i porównaj z teorią) położenie miejsc zerowych.

2.2 Widmo sygnału prostokątnego

Skrypt Zad2_2_sygnal_prostokatny.

- Wykonaj skrypt i przedyskutuj wykresy tworzone w kolejnych etapach.

2.3 Współczynnik wypełnienia(*)

Skrypt Zad2_3_wspolczynnik_wypelnienia.

- Zbadaj jakie harmoniczne częstotliwości sygnału $F_{\text{sig}} = 1/T$ widać w widmie dla różnych współczynników wypełnienia K/T , zmieniając K jako:
 - $K = 25$
 - $K = 24$
 - $K = 21$
 - $K = 20$
 - $K = 15$
- Czy potrafisz uzasadnić kształt widma?

Raport

1. Przedstaw wykresy dla $K=25$ (współczynnik wypełnienia 50%) oraz $K=21$.
2. Podaj wzór (wzory) i oblicz położenia teoretyczne pierwszych 10. harmonicznych sygnału prostokątnego.
3. Podaj położenia (indeksy) wszystkich niezerowych prążków w widmie sygnału prostokątnego w zakresie indeksów $[0-40]$
4. Podaj które harmoniczne sygnału obserwujemy w obu przypadkach. Uzasadnij to zachowanie teoretycznie (możesz oczywiście odnieść się do poprzedniego zadania)

3 Wyciek DFT

Skrypt Zad3_wyciek.

- Wykonaj skrypt dla $F_{\text{sig}} = 2$ kHz. Przedyskutuj wyniki.
- Powtórz skrypt dla $F_{\text{sig}} = 2.5$ kHz. Przedyskutuj wyniki - która z interpretacji powstawania skwantowanego widma jest prawidłowa?

4 Widmo instrumentów muzycznych

Skrypt Zad4_widmo_instrumentow.

- Wykonaj skrypt i przedyskutuj wyniki.

5 Okna DFT

Skrypt Zad5_okna.

- Wykonaj skrypt dla $\text{Amp1} = 0$. Przedyskutuj wykresy - zwróć szczególną uwagę to, jaką częstotliwość sygnału (w porównaniu z zadaną F_{sig2}) możemy odczytać z wykresów widm bez i z różnymi funkcjami okien.

- Przedyskutuj wpływ różnych typów okna na kształt wycieku DFT.
- Ustaw $\text{Amp1} = 0.01$. Doda to drugą składową o częstotliwości 3 kHz, ale bardzo małej amplitudzie. Odpowiedz na pytania:
 - Czy w widmie bez użycia okien widzisz pik sygnału 1?
 - Czy zastosowanie okien poprawia sytuację? Które z okien dało najlepsze rezultaty?

6 Rozdzielczość transformaty DFT

6.1 Rozdzielczość a wyciek(*)

Skrypt `Zad6_1_rozdzielczosc_wyciek`.

- Wykonaj skrypt. Czy w widmie widzisz pik dla $F_{\text{sig1}} = 3$ kHz?
- Zwiększaj `zeroPadding` (logarytmicznie 10, 100, 1000...):
 - Przy jakiej wartości "Polozenie maksimum widma" zaczyna odpowiadać rzeczywistej wartości F_{sig2} ?
 - Czy `zero padding` pomaga wyłuskać częstotliwość F_{sig1} z widma?
- Zmień z powrotem `zeroPadding` na zero. Zwiększ `N` do 2400, 3200. Zaobserwuj zmiany widoczności piku dla F_{sig1} .
- Zwiększ `N` do 16000. Zaobserwuj zmianę widma.

Raport

1. Przedstaw wartość `zeroPadding` dla której położenie maksimum widma zaczyna odpowiadać zadanej wartości $F_{\text{sig2}} = 3047$ Hz.
2. Przedstaw wykres dla tej wartości `zeroPadding`.
3. Skomentuj działanie `zero padding` - dlaczego pomaga on lepiej zlokalizować pik dla częstotliwości powodującej wyciek DFT?
4. Skomentuj czy `zero padding` pomógł wyłuskać częstotliwość F_{sig1} z widma? Dlaczego się tak dzieje?
5. Przedstaw wykresy dla `N = 2400` oraz `3200`. Skomentuj widoczność piku 3 kHz w obu przypadkach, a dokładnie: dlaczego zwiększanie `N` w ogólności powinno pomóc? Dlaczego nie działa dla `N = 3200`? (Podpowiedź: oblicz stosunek $F_{\text{sig2}}/F_{\text{base}}$ dla obu wartości `N`. Jakiego wycieku spodziewamy się w obu przypadkach? W którym z nich mamy większą szansę zaobserwować F_{sig1} ?)

6.2 Rozdzielczość a szum

Skrypt `Zad_6_2_rozdzielczosc_szum`.

Wykonaj skrypt i przedyskutuj wyniki. Dlaczego pik sygnału jest coraz to bardziej widoczny wraz ze wzrostem `N`?

7 Transformata odwrotna – filtracja w dziedzinie częstotliwości

7.1 Filtracja sygnału prostokątnego(*)

Skrypt `Zad7_1_transformata_odwrotna_prostokat.`

Wykonaj skrypt. Przedyskutuj przebieg rekonstrukcji w obu przypadkach (`filterType = 1` oraz `2`). Spróbuj odpowiedzieć na pytanie jakie filtry efektywnie implementujemy w obu przypadkach?

Raport

1. Przedstaw po jednym wykresie dla wybranego kroku rekonstrukcji dla obu przypadków (uwaga - proszę NIE wybierać efektu końcowego!)
2. Opisz jakie teoretycznie filtry implementujemy i jak wpływa to na kształt zrekonstruowanego przebiegu w wybranym kroku.

7.2 Filtracja muzyki

Skrypt `Zad7_2_muzyka_filtracja.`

- Wykonaj skrypt dla wszystkich trzech typów filtrów. Możesz również pozmienić ich częstotliwości graniczne i odsłuchać jaki efekt mają na filtrowaną muzykę.
- Opcjonalnie w domu - zastosuj skrypt do ulubionej muzyki (nie możemy zrobić tego na zajęciach ze względów licencyjnych)

7.3 Filtracja przydźwięku sieciowego

Skrypt `Zad7_3_filtracja_przydzwiek.`

- Wykonaj skrypt dla pliku „media/2L-092_01_44kHz_32b_50Hz_hum.mp3”.
 - Oceń skuteczność filtracji 50Hz oraz 50Hz z harmonicznymi;
 - Dobierz parametr `filterWidth` tak, aby uzyskać najlepszy efekt filtracji z harmonicznymi z minimalnymi możliwym zniekształceniami.
- Powtórz zadanie dla pliku „media/2L-092_01_44kHz_32b_50Hz_hum_real.mp3”. Czy dla bardziej złożonego przydźwięku nasza prosta filtracja dalej działa?

8 Transformata DFT obrazu

8.1 Podstawy DFT obrazu – pionowe i poziome linie

Skrypt `Zad8_1_dft_obraz_podstawy.`

Wykonaj skrypt i przedyskutuj wykresy.

8.2 Podstawy DFT obrazu – animacja skośnych linii

Skrypt `Zad8_2_dft_obraz_podstawy_2`.

Wykonaj skrypt i przedyskutuj wykresy.

Zwróć uwagę że w tym zadaniu generujemy widmo w postaci dwóch punktów, a następnie z widma, poprzez transformatę odwrotną, tworzymy obrazek.

8.3 Podstawy DFT obrazu – obrazy bardziej złożone(*)

Skrypt `Zad8_3_dft_obraz_podstawy_3`.

Wykonaj kolejne kroki skryptu i przedyskutuj widma. W przeciwieństwie do poprzedniego zadania, tutaj (Figure 4) najpierw tworzymy obrazek skośnych linii (obracając o zadany kąt pionowe linie), a potem dokonujemy jego analizy częstotliwościowej (Figure 5 oraz 6). Aby zrozumieć te widma, przeanalizuj wykres z Figure 7, prezentujący jeden rząd pikseli obrazka z Figure 4.

Raport

1. Przedstaw rysunek 1 (pionowe i poziome linie w dwóch połówkach obrazka) i jego widmo w skalach liniowej i logarytmicznej. Przedstaw interpretację widma logarytmicznego.

8.4 Podstawy DFT obrazu – widmo okręgu

Skrypt `Zad8_4_dft_obraz_podstawy_4`.

Wykonaj skrypt i przedyskutuj wyniki.

8.5 DFT realnych zdjęć

Skrypt `Zad8_5_dft_zdjec`.

Wykonaj skrypt i przedyskutuj wyniki.

9 Filtracja zdjęć w dziedzinie częstotliwości

9.1 Filtracja dolnoprzepustowa zdjęć(*)

Skrypt `Zad9_1_filtracja_dolnoprzepustowa_zdjec`.

- Wykonaj skrypt dla `filterType = 1`. Przedyskutuj wyniki (widmo i przefiltrowany obraz).
- Sprawdź działanie pozostałych filtrów.
- Dla filtru Butterworth-a zmień rząd filtru (na np. 20) i przedyskutuj różnice.

Raport

1. Umieść przefiltrowane obrazy dla filtrów typu 1 i 3 (koło i Butterworth rzędu 2).
2. Omów jaki efekt na zdjęcie ma zastosowanie filtru dolnoprzepustowego i dlacego.
3. Podaj jaki charakterystyczny efekt widać dla zdjęcia przefiltrowanego z użyciem filtru typu 1. Zaznacz na zdjęciu jeden obszar w którym widać efekt. Uzasadnij (zadanie 8.4) źródło efektu.
4. Skomentuj dlacego efektu nie widać dla filtru Butterworth-a rzędu drugiego, a da się go zauważyć dla rzędu 20.?

9.2 Filtracja górnoprzepustowa zdjęć

Skrypt `Zad9_2_filtracja_gornoprzepustowa_zdjec.`

- Wykonaj skrypt dla `filterType = 1`. Przedyskutuj wyniki (widmo i przefiltrowany obraz).
- Powtórz skrypt dla parametru `keepDC = 1`.
- Sprawdź działanie pozostałych filtrów.
- Dla filtru Butterworth-a zmień rząd filtru (na np. 20) i przedyskutuj różnice.

9.3 Filtracja pasmowa zdjęć

Skrypt `Zad9_3_filtracja_pasmowa_zdjec.`

Wykonaj skrypt i przedyskutuj wyniki.

9.4 Operatory Sobel'a

Skrypt `Zad9_4_operatory_Sobela.`

Wykonaj skrypt i przedyskutuj wyniki.

9.5 Prosty algorytm wyostrzający(*)

Skrypt `Zad9_5_wyostrzenie_zdjec.`

Wykonaj skrypt. Spróbuj, sterując parametrami `filterRatio`, `butterFilterOrder` i `sharpeningCoef` uzyskać jak najlepszy efekt wyostrzenia zdjęcia.

UWAGA: algorytm jest prymitywny, więc nie należy spodziewać się fenomenalnych rezultatów.

Raport

1. Przedstaw wartości parametrów dających, Twoim zdaniem, najlepsze efekty. Przedstaw wyostrzone zdjęcie dla tego zestawu.

10 Kompresja dźwięku i obrazu

10.1 Kompresja „losowa”

Skrypt Zad10_1_kompresja_losowa.

- Wykonaj skrypt zmniejszając stopniowo parametr quality. Zaobserwuj zmiany współczynnika kompresji.
- Czy jakakolwiek wartość parametru quality zapewnia akceptowalny Twoim zdaniem rezultat?

10.2 MDCT – spektrogram(*)

Skrypt Zad10_2_spektrogram.

- Wykonaj skrypt dla pliku dialog.mp3. Przedyskutuj wynik, czyli to, co przedstawia spektrogram.
- Wykonaj skalowanie spektrogramu - wyraż oś OX w [s], a OY w [Hz].
- Powtórz zadanie dla pliku piano_scale.mp3. Dlaczego pod koniec spektrogramu nie możemy rozróżnić pojedynczych dźwięków?
- Zmień frameSize na 16384. Czy pomogło to na problem 3.? Co się jeszcze zmieniło?
- Wykonaj spektrogram dla pliku 2L-092_01_44kHz_32b.mp3. Jak zmienia się spektrogram dla frameSize = 1024 i 16384?
Podpowiedź: zakomentuj `close all` w pierwszej linii skryptu, wykonaj skrypt dwa razy i porównaj spektrogramy.
- Po wykonaniu zadania z kompresji mowy wykonaj spektrogramy słowa „strong” wymawianego przez kobietę i mężczyznę, odkomentowując linie 28–31

Raport

1. Przedstaw spektrogram dla pliku dialog.mp3 wyrażony w [s] i [Hz] i opisz co przedstawia.
2. Skomentuj jak wpływa na rozdzielczość częstotliwościową, a jak na czasową, zwiększenie długości ramki MDCT (frameSize).

10.3 Kompresja mowy

Skrypt Zad10_3_kompresja_mowy.

- Wykonaj skrypt dla różnych wartości parametru quality (np. 0.999, 0.99 oraz 0.9). Przedyskutuj wyniki.
- Ustal wartość parametru quality dającej akceptowalny Twoim zdaniem rezultat.
- Odkomentuj linie 12–15 i wykonaj kompresję słowa „strong” wymawianego przez kobietę (her) i mężczyznę (him). Zwróć uwagę na uzyskany stopień kompresji

w obu przypadkach. Aby zrozumieć różnicę wykonaj, z pomocą poprzedniego skryptu, spektrogramy dla obu przypadków.

- Ponów punkt 2. dla pozostałych plików audio.

10.4 Kompresja obrazu – JPEG(*)

Skrypt Zad10_4_kompresja_JPEG.

- Wykonaj skrypt zmniejszając stopniowo parametr quality. Zaobserwuj zmiany współczynnika kompresji.
- Znajdź wartość parametru quality zapewniającą akceptowalny Twoim zdaniem rezultat.

Raport

1. Przedstaw wartość parametru quality dającego akceptowalny Twoim zdaniem rezultat, oraz uzyskany stopień kompresji.
2. Przedstaw skompresowany obraz dla tej wartości quality i opisz zmiany w stosunku do oryginału jakie widzisz.
3. Zaprezentuj działanie kompresji JPEG na przykładzie jednego bloku:
 - a) Wypisz i zamieść w sprawozdaniu jeden, wybrany blok 8x8 pikseli, po transformacji DCT i jej kwantyzacji, np: `dctImage(1:8, 1:8)` oraz `dctQuantImage(1:8, 1:8)` (możesz wybrać dowolny inny blok, ale indeksy muszą się zmieniać co 8, czyli np. 57:64, ale nie 53:60)
 - b) Porównaj ze sobą dwa bloki i skomentuj - co robi kompresja stratna JPEG, w jaki sposób redukuje ilość danych (rozmiar pliku ze zdjęciem) i dlaczego wpływa na jakość obrazu? Jakiej, w przybliżeniu, filtracji odpowiada?
4. Uzasadnij źródło zmian skompresowanego obrazu w stosunku do oryginału w odniesieniu do opisu działania kompresji JPEG.

10.5 Kompresja bezstratna – kodowanie Huffman'a(*)

To zadanie nie ma skryptu w Matlab-ie.

Kodowanie Huffman'a jest jedną z metod kodowania entropijnego – zależnego od częstości wystąpienia danego symbolu (np. litery, liczby) w ciągu danych. Im częściej dany symbol występuje, tym krótszy kod (np. mający mniej bitów) staramy się mu przypisać.

Diagram kodowania można wykonać dla zdania, traktując pojedyncze znaki jako symbole. Dla przykładu rozważmy zdanie „ALA MA LAMPARTA”.

W pierwszej kolejności należy zidentyfikować unikalne symbole (litery) i policzyć ilość ich wystąpień. Formalnie powinniśmy przedstawić prawdopodobieństwo ich wystąpienia, ale w tym przypadku sprowadza się to do podzielenia każdej ilości wystąpień przez

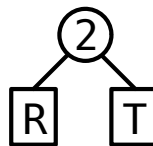
całkowitą długość ciągu znaków (15), więc dla uproszczenia opisu dzielenie to zostanie pominięte.

W wyniku analizy zdania „ALA MA LAMPARTA” można uzyskać następującą tabelę unikatowych liter i ilości ich wystąpień:

Znak	Ilość wystąpień
A	6
L	2
spacja: _	2
M	2
P	1
R	1
T	1

Następnie zaczynamy budować drzewo Huffman’a, łącząc dwa elementy o najmniejszym prawdopodobieństwie wystąpienia węzłem, w który wpisujemy sumę prawdopodobieństw obu połączonych elementów. W rozważanym przypadku są trzy elementy o najmniejszym prawdopodobieństwie 1: litery P, R, T. Nie ma żadnej reguły, które dwa z nich należy wybrać – wybór jest dowolny i każdy z nich da równie optymalne kodowanie. Z tego względu dla jednego ciągu danych można zbudować wiele różnych, ale równie dobrych drzew.

W przykładzie połączmy dwie ostatnie litery – R oraz T. Początek drzewa będzie więc wyglądał jak na rysunku 1.

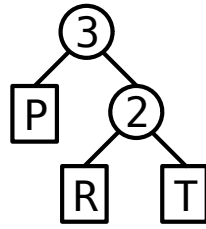


Rysunek 1: Początek drzewa Huffman’a

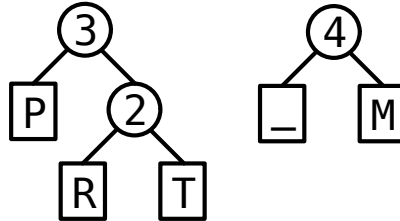
W następnej kolejności zostaje jeden element o prawdopodobieństwie 1: litera P, oraz **cztery** elementy o prawdopodobieństwie 2: trzy znaki L, M, spacja, oraz stworzony przed chwilą węzeł. Musimy wybrać literę P (ma najniższe prawdopodobieństwo), oraz, dowolnie, jeden z czterech pozostałych elementów. Wybierzmy węzeł, co da drzewo postaci jak na rysunku 2.

W kolejnym kroku musimy wybrać dwa z pozostałych trzech elementów o prawdopodobieństwie 2. Wybierzmy spację oraz M – rysunek 3.

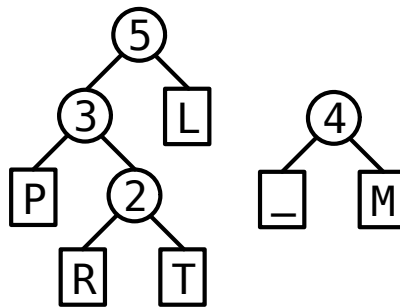
Następnie pozostaje znak L o prawdopodobieństwie 2, zaś kolejne najniższe prawdopodobieństwo 3 ma jeden z węzłów drzewa. Łączymy więc oba elementy jak na rysunku 4. W piątym kroku widać, że dwa węzły drzewa mają niższe prawdopodobieństwo (odpowiednio 5 oraz 4) niż jedyna, nieuwzględniona jeszcze w drzewie, litera A. Łączymy więc najpierw dwa węzły w jeden, o prawdopodobieństwie 9, a następnie powstały węzeł z literą A. Otrzymujemy finalny węzeł o prawdopodobieństwie 15 (czyli dokładnie tyle, ile mamy znaków w zdaniu). Ponieważ nie mamy już więcej znaków, zaś drzewo



Rysunek 2: Drzewo Huffman'a w drugim kroku



Rysunek 3: Drzewo Huffman'a w trzecim kroku



Rysunek 4: Drzewo Huffman'a w czwartym kroku

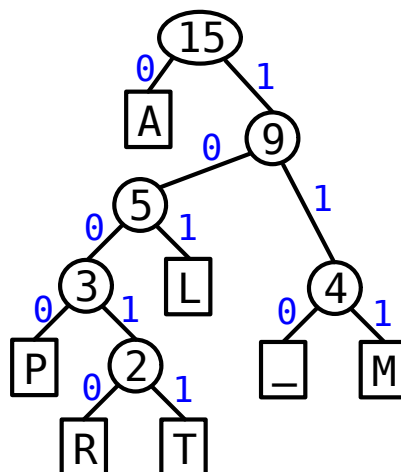
zaczyna się pojedynczym węzłem (tzw. korzeń), zakończyliśmy budowę drzewa. W ostatnim kroku każdemu z rozgałęzień w lewo nadajemy wartość bitową 0, a rozgałęzieniu w prawo – 1. Daje to finalnie drzewo przedstawione na rysunku 5

Aby otrzymać kod odpowiadający danemu znakowi, poruszamy się po drzewie zaczynając od korzenia, do liścia zawierającego dany znak, notując po kolei wartości bitów odgałęzień po których się poruszamy.

Poniżej pokazano przykładowe kody znaków:

- A – 0
- L – 1 0 1
- T – 1 0 0 1 1

Widać stąd, że im rzadziej dany znak występuje w zdaniu, tym dłuższy ciąg bitów mu



Rysunek 5: Finalna postać drzewa Huffman'a

odpowiada.

Znając kody liter można zapisać całe zdanie „ALA MA LAMPARTA” jako następujący ciąg bitów:

0 101 0 110 111 0 110 101 0 111 1000 0 10010 10011 0

Spacje pomiędzy ciągi bitowe reprezentujące poszczególne znaki zostały wstawione sztucznie – w rzeczywistości ciąg bitów będzie ich pozbawiony. Ponieważ ilość bitów na znak jest zmienna, powstaje pytanie jak dekodery, którego zadaniem jest odtworzyć (zdekompresować) wiadomość ma wiedzieć, który bit należy do którego znaku?

Aby odpowiedzieć na to pytanie, spróbujemy rozkodować ten ciąg z punktu widzenia dekodera. Algorytm jest następujący: bierzemy pierwszy bit i, zaczynając od korzenia, wybieramy odpowiednie odgałęzienie. Jeżeli napotykamy węzeł, pobieramy z ciągu kolejny bit i idziemy dalej. Jeżeli napotykamy liść, wiemy że znaleźliśmy znak (z liścia drzewa nie ma gdzie iść dalej), więc zapisujemy go do zdekodowanej informacji. Pobieramy następny bit zakodowanego ciągu i, ponieważ w poprzednim kroku znaleźliśmy liść, rozpoczynamy podróż po drzewie znów od korzenia, czyli wracamy do początku tego algorytmu. Tym samym automatycznie wykrywamy ilość bitów potrzebnych na zdekodowanie danego znaku.

W przypadku rozważanej wiadomości początek dekodowania przebiegać będzie następująco (**wytłuszczoną** czcionką zaznaczono aktualnie analizowany ciąg bitów, *pochyłą* – bitu już „zużyte”)

1. **0**10101101110110101111000010010100110 – pobieramy 0, z korzenia prowadzi nas ono bezpośrednio do liścia A – zapisujemy znak do wiadomości.
2. 0 **1**010110111011010101111000010010100110 – pobieramy kolejny bit – 1. Ponieważ w poprzednim kroku znaleźliśmy liść, zaczynamy podróż przez drzewo od początku. Bit 1 kieruje nas do węzła o prawdopodobieństwie 9. Ponieważ jest to

węzeł, idziemy dalej;

3. 0 1010110111011010101111000010010100110 – pobieramy kolejny bit – 0. Z węzła o prawdopodobieństwie 9 prowadzi on do węzła o prawdopodobieństwie 5. Ponieważ jest to węzeł, idziemy dalej;
4. 0 1010110111011010101111000010010100110 – pobieramy kolejny bit – 1. Z węzła o prawdopodobieństwie 5 prowadzi on do liścia L. Nasza wiadomość to, na razie, „AL”.
5. 0 101 0110111011010101111000010010100110 – pobieramy kolejny bit – 0. Ponieważ w poprzednim kroku znaleźliśmy liść, zaczynamy podróż przez drzewo od początku. Bit prowadzi nas ono bezpośrednio do liścia A – zapisujemy znak do wiadomości: „ALA”.
6. 0 101 0 110111011010101111000010010100110 – pobieramy kolejny bit – 1. Ponieważ w poprzednim kroku znaleźliśmy liść, zaczynamy podróż przez drzewo od początku. Bit 1 kieruje nas do węzła o prawdopodobieństwie 9. Ponieważ jest to węzeł, idziemy dalej... itd.

Policzmy efektywność algorytmu – uzyskany stopień kompresji. Gdyby zdanie zapisać z użyciem kodowania ASCII lub UTF-8, potrzebne byłoby $15 \cdot 8 = 120$ bitów. Ciąg uzyskany z kodowania Huffman’a ma długość 38 bitów. Skompresowaliśmy więc ciąg $\frac{120}{38} \sim 3.16$ raza. Niestety, ponieważ dla takiego samego ciągu można uzyskać różne drzewa Huffman’a, drzewo którego użyliśmy w tym konkretnym przypadku należy również zapisać w skompresowanym pliku... Dla tak krótkiego ciągu znaków ilość danych potrzebnych do zapisania drzewa przewyższy zysk z kodowania (choć samo drzewo też można sprytnie skompresować, ale jest to już poza granicami tego opracowania), ale dla odpowiednio długiego ciągu danych nadal uzyskamy spory zysk.

Efektywność algorytmu można też przedstawić licząc średnią ilość bitów na znak. Można to zrobić trywialnie jako stosunek długości zakodowanego ciągu do ilości znaków: $\frac{38}{15} = 2.5(3)$, o ile mamy już zakodowaną wiadomość.

Średnią ilość bitów na znak można też wyliczyć samego drzewa, bez potrzeby kodowania całej wiadomości (i zliczania jej długości) jako:

$$H = \sum_x p(x) \cdot L(x) \quad (2)$$

gdzie:

- $p(x)$ to prawdopodobieństwo wystąpienia symbolu (znaku) x ;
- $L(x)$ to długość (ilość bitów) kodu odpowiadającego symbolowi x .

W naszym przypadku będzie to więc:

$$H = \frac{6}{15} \cdot 1 \left|_A + \frac{2}{15} \cdot 3 \left|_L + \frac{2}{15} \cdot 3 \left|_{_} + \frac{2}{15} \cdot 3 \left|_M + \frac{1}{15} \cdot 4 \left|_P + \frac{1}{15} \cdot 5 \left|_R + \frac{1}{15} \cdot 5 \left|_T = 2.5(3) \quad (3)$$

Idealną (minimalną możliwą) długość słowa kodowego dla danego ciągu można obliczyć

z podstawowego twierdzenia Shannona (Shannon's source coding theorem) jako:

$$H_{ideal} = - \sum_x p(x) \cdot \log_2[p(x)] \quad (4)$$

Dla naszego ciągu dostaniemy więc

$$H_{ideal} = -\frac{6}{15} \cdot \log_2 \left(\frac{6}{15} \right) \Big|_A - \frac{2}{15} \cdot \log_2 \left(\frac{2}{15} \right) \Big|_L - \dots = 2.473 \quad (5)$$

Widać więc że kodowanie Huffman'a daje, pomimo relatywnej prostoty algorytmu, niemalże idealnie krótki ciąg kodowy ($\frac{2.5(3)}{2.473} = 1.024$, czyli zaledwie 2.4% więcej, niż najkrótszy teoretycznie możliwy).

Raport

1. Zbuduj drzewo Huffman'a ciągu „FILTR CYFROWY FIR”, albo „NA BEZRYBIU I RAK RYBA”. Opcjonalnie wybierz własne zdanie (co najmniej 15 znaków), postaraj się jednak aby zawierało ono powtarzające się litery.
 - a) Podaj zdanie które kodujesz, policz i podaj jego długość (ilość znaków, pamiętaj o spacjach);
 - b) Zbuduj i przedstaw tabelę unikatowych symboli i ich prawdopodobieństw (ilości wystąpień) jak w przykładzie;
 - c) Narysuj i przedstaw drzewo Huffman'a (może być np. zdjęcie odręcznego rysunku, byle w miarę czytelne);
 - d) Zakoduj zdanie i przedstaw ciąg bitowy który go reprezentuje;
 - e) Oblicz średnią ilość bitów na znak H dla otrzymanego kodu oraz jej wartość idealną H_{ideal} i oceń skuteczność kodu.