# Rozdział 1

# Odległości na przestrzeni ciągów znaków

## 1.1. Podstawowe definicje

CZY TO JAKO CIĄGŁY TEKST CZY WSZYSTKO PISAĆ W DEFINICJI???

Niech $\Sigma = \{\Sigma_i\}$ będzie skończonym uporządkowanym alfabetem o wielkości $|\Sigma|$. *Napisem* nazywamy skończony ciąg znaków z $\Sigma$. Zbiór wszystkich napisów o długości $n$ nad $\Sigma$ jest oznaczony przez $\Sigma^n$, podczas gdy przez $\Sigma^* = \bigcup_{n=1}^{\infty} \Sigma^n$ rozumiemy zbiór wszystkich napisów utworzonych ze znaków z $\Sigma$ [1].

O ile nie podano inaczej, używamy zmiennych $s$, $t$, $u$, $v$, $w$, $x$, $y$ jako oznaczenie napisów oraz $a$, $b$, $c$ do oznaczenia napisów jednoznakowych albo po prostu *znaków*. Pusty napis jest oznaczany przez $\varepsilon$. Przez $|s|$, dla każdego napisu $s \in \Sigma^*$, rozumiemy jego długość, czyli liczbę znaków w napisie. Ciąg zmiennych oznaczających napisy i/lub znaki oznaczają ich złączenie [1]. Dla rozróżnienia napisów od zmiennych reprezentujących napis, te pierwsze oznaczamy pismem maszynowym, np. `napis`.

Poprzez $s_i$ rozumiemy $i$-ty znak z napisu $s$, dla każdego $i \in \{1, \ldots, |s|\}$. Podciąg kolejnych przylegających do siebie znaków z napisu nazywamy *podnapisem*. Podnapisem napisu $s$, który zaczyna się od $i$-tego znaku, a kończy na $j$-tym znaku, oznaczamy przez $s_{i:j}$, tj. $s_{i:j} = s_i s_{i+1} \ldots s_j$ dla $i < j$. Zakładamy również, że jeśli $j < i$, to $s_{i:j} = \varepsilon$ [1, 7].

Załóżmy, że napis $s$ jest resprezentacją złączenia trzech, być może pustych, podnapisów $w$, $x$ i $y$, tj. $s = wxy$. Wówczas podnapis $w$ nazywamy *prefiksem*, natomiast podnapis $y$ – sufiksem [1].

Podnapis złożony z kolejnych znaków napisu, o ustalonej długości $q$ jest nazywany *$q$-gramem*. $q$-gramy o $q$ równym jeden, dwa lub trzy mają specjalne nazwy: *unigram, bigram* i *trigram*. Jeśli $q > |s|$, to $q$-gramy napisu $s$ są napisami pustymi [1].

**Przykład 1.1.** Niech $\Sigma$ będzie alfabetem złożonym z 26 małych liter alfabetu łacińskiego oraz niech $s =$ `ela`. Wówczas mamy $|s| = 3$, $s \in \Sigma^3$ oraz $s \in \Sigma$. Co więcej, mamy $s_1 =$ `e`, $s_2 =$ `l`, $s_3 =$ `a`. Podnapis $1 : 2$ napisu $s$ to $s_{1:2} =$ `el`. W napisie tym mamy do czynienia jedynie z $q$-gramami o $q$ równym jeden, dwa oraz trzy: `e`, `l`, `a`; `el`, `la` oraz `ela` odpowiednio.

## 1.2. Odległości na przestrzeni ciągów znaków

W tym podrozdziale zajmiemy się odległościami na przestrzeni ciągów znaków. Można je podzielić na trzy grupy:

- oparte na operacjach edycyjnych (*edit operations*),
- oparte na $q$-gramach,
- miary heurystyczne.

BLA BLA JAKIEŚ LANIE WODY O METRYKACH Pierwszy rodzaj odległości jest najczęściej używany w algorytmach zajmujących się optymalnym dopasowaniem, dlatego też poświęcimy mu największą część niniejszego rozdziału. Odległości oparte na $q$-gramach ... (TUTAJ COŚ O NICH). Natomiast miary heurystyczne są rzadko stosowane, będąc zazwyczaj używane w konkretnych przypadkach. Miary te miały jednak swój wkład w historię optymalizacji napisów, zatem pokrótce opiszemy je pod koniec tego rozdziału.

### 1.2.1. Odległości oparte na operacjach edycyjnych

HISTORIA ODLEGLOSCI EDYCYJNYCH?

**Ścieżka edycyjna i bazowe operacje edycyjne.** *Odległość edycyjna* $ED(s,t)$ pomiędzy dwoma napisami $s$ i $t$ to minimalna liczba operacji edycyjnych potrzebna do przetworzenia $s$ w $t$ (i $\infty$, gdy taki ciąg nie istnieje) [4]. *Ścisłą odległością edycyjną* nazywamy minimalną liczbę nie nakładających się operacji edycyjnych, które pozwalają przekształcić jeden napis w drugi, i które nie przekształcają dwa razy tego samego podnapisu [1].

Napis może zostać przetworzony w drugi poprzez wykonanie na nim ciągu przekształceń jego podnapisów. Ten ciąg nazywany jest *ścieżką edycyjną (śladem edycji?)*, podczas gdy przekształcenia są nazywane *bazowymi operacjami edycyjnymi*. Bazowe operacje edycyjne, które polegają na mapowaniu napisu $s$ w napis $t$, są oznaczane przez $s \to t$. Zbiór wszystkich bazowych operacji edycyjnych oznaczamy przez $\mathbb{B}$ [1].

Bazowe operacje edycyjne są zazwyczaj ograniczone do:

- usunięcie znaku: `l` $\to \varepsilon$, tj. usunięcie litery `l` , np. `ela` $\to$ `ea`,
- wstawienie znaku: $\varepsilon \to$ `k`, tj. wstawienie litery `k`, np. `ela` $\to$ `elka`,
- zamiana znaku: `e` $\to$ `a`, tj. zamiana litery `e` na `a`, np. `ala` $\to$ `ela`,
- transpozycja: `el` $\to$ `le`, tj. przestawienie dwóch przylegających liter `e` i `l`, np. `ela` $\to$ `lea`.

**Własność 1.1.** *Zakładamy, że* $\mathbb{B}$ *spełnia następujące własności [1]:*

- *jeśli* $s \to t \in \mathbb{B}$*, to odwrotna operacja* $t \to s$ *również należy do* $\mathbb{B}$*;*
- $a \to a \in \mathbb{B}$ *(operacja identycznościowa dla jednego znaku należy do* $\mathbb{B}$*);*
- *zbiór* $\mathbb{B}$ *jest zupełny: dla dwóch dowolnych napisów* $s$ *i* $t$*, istnieje ślad edycji, który przekształca* $s$ *w* $t$*.*

Zauważmy, że zbiór $\mathbb{B}$ nie musi być skończony.

**Odległość edycyjna.** Podobieństwo dwóch napisów może być wyrażone jako długość ścieżki edycyjnej, dzięki której jeden napis zostaje przekształcony w drugi:

**Definicja 1.1.** *Mając dany zbiór bazowych operacji edycyjnych,* odległość edycyjna $ED(s,t)$ *jest równa długości najkrótszej ścieżki edycyjnej, która przekształca napis $s$ w napis $t$. Najkrótsza ścieżka, która przekształca napis $s$ w napis $t$ jest nazywana* optymalną *ścieżką edycyjną [1].*

Przykładowe odległości edycyjne: Hamminga, najdłuższego wspólnego podnapisu (*longest common substring*), Levenshteina, optymalnego dopasowania napisów (*optimal string alignment*), Damareu-Levenshteina. Odległości te różnią się zbiorem bazowych operacji edycyjnych. Jeśli w zbiorze tym znajduje się tylko zamiana znaków, to mamy do czynienia z odległością Hamminga. Gdy zbiór bazowych operacji edycyjnych zawiera wstawienia i usunięcia znaków, to jest to odległość najdłuższego wspólnego podnapisu. Gdyby $\mathbb{B}$ powiększyć o zamianę znaków, to otrzymamy odległość Levenshteina. Dwie ostatnie odległości, tj. optymalnego dopasowania napisów oraz Damareu-Levenshteina, mają w zbiorze bazowych operacji edycyjnych usunięcie, wstawienie, zamianę oraz transpozycję znaków.

**Przykład 1.2.** JAKIŚ PRZYKŁĄD SCIEZKI I OPTYMALNEJ SCIEZKI.

Definicja odległości edycyjnej może być również interpretowana jako minimalny koszt, dzięki któremu przekształcamy jeden napis w drugi. Definicję można uogólnić na dwa sposoby. Po pierwsze, bazowe operacje edycyjne mogą mieć przydzielone koszty (wagi) $\delta(a \to b)$ [8]. Zazwyczaj koszt każdej operacji wynosi jeden, jednak można, na przykład, nadać transpozycji mniejszy koszt niż operacji wstawienia znaku. Dalej, można rozszerzyć funkcję kosztu $\delta$ na ścieżkę edycyjną $E = a_1 \to b_1, a_2 \to b_2, \ldots, a_{E|} \to b_{|E|}$ poprzez $\delta(E) = \sum_{i=1}^{|E|} \delta(a_i \to b_i)$ [1]. Odtąd poprzez odległość między napisem $s$ a napisem $t$ będziemy rozumieć minimalny ze wszystkich możliwych kosztów ścieżek przekształcających $s$ w $t$. Odległości zdefiniowane w ten sposób zazwyczaj są nazwyane *uogólnionymi* odległościami edycyjnymi.
Po drugie, zbiór operacji edycyjnych $\mathbb{B}$ może zostać rozszerzony o ważone zamiany (substytucje) (pod)napisów, zamiast operacji edycyjnych wykonywanych na pojedynczych znakach [6]. Odległości zdefiniowane w ten sposób zazwyczaj są nazwyane *rozszerzonymi* odległościami edycyjnymi. Przykładowo, $\mathbb{B}$ może zawierać operację `x` $\to$ `ks` o koszcie jednostkowym. Wówczas rozszerzona odległość pomiędzy napisami `xero` i `ksero` wynosi jeden, podczas gdy standardowa (zwykła, nierozszerzona) odległość wyniosłaby dwa [1].

**Definicja 1.2.** *Mając dany zbiór bazowych operacji edycyjnych $\mathbb{B}$ oraz funkcję $\delta$, która nadaje koszt wszystkim bazowym operacjom edycyjnym z $\mathbb{B}$, ogólna(?) odległość edycyjna pomiędzy napisami $s$ i $t$ jest zdefiniowana jako minimalny koszt ścieżki edycyjnej, która przekształca $s$ w $t$ [1].*

Zazwyczaj koszt pojedynczej operacji operacji z $\mathbb{B}$ jest równy jeden. Czasem jednak nadaje się poszczególnym operacjom różne koszty, dając np. transpozycji mniejszą wagę niż wstawieniu znaku. Gdy koszt wszystkich operacji jest równy jeden, to mówimy po prostu o odległości edycyjnej, natomiast gdy różne operacje mają różne wagi, to mówimy o *uogólnionej* odległości edycyjnej.

**Własność 1.2.** *Zakładamy, że funkcja kosztu $\delta(s \to t)$ ma następujące własności [1]:*

- $\delta(s \to t) \in \mathbb{R}$ *(koszt operacji jest liczbą rzeczywistą),*
- $\delta(s \to t) = \delta(t \to s)$ *(symetria),*
- $\delta(s \to t) \geq 0,\ \delta(s \to s) = 0$ *i* $\delta(s \to t) = 0 \Rightarrow s = t$ *(dodatnia określoność ??),*
- $\forall \gamma > 0$ *zbiór bazowych operacji* $\{s \to t \in \mathbb{B} | \delta(s \to t) < \gamma\}$ *jest skończony (skończoność podzbioru bazowych operacji, których koszt jest ograniczony z góry).*

Zauważmy, że ostatnia własność jest zawsze spełniona dla skończonego zbioru $\mathbb{B}$.

**Twierdzenie 1.3.** *Z własności 1.1 i 1.2 wynika, że:*

- *dla każdych dwóch napisów $s$ i $t$, istnieje ścieżka o minimalnym koszcie, tj. właściwie (prawdziwie, odpowiednio?) zdefiniowana odległość edycyjna z $s$ do $t$ [1],*
- *ogólna odległość edycyjna z definicji 1.2 jest metryką [8].*

*Dowód.* Żeby udowodnić, że $ED(s,t)$ jest metryką, musimy pokazać, że $ED(s,t)$ istnieje, jest dodatnio określona, symetryczna oraz subaddytywna (tj. spełnia nierówność trójkąta).

Z własności 1.2 wynika, że funkcja kosztu jest nieujemna (JA TEGO NIW WIDZE) i że tylko identyczność ma koszt równy zero. Stąd, bez utraty ogólności, możemy rozważyć jedynie takie ścieżki edycyjne, które nie zawierają operacji identycznościowych. Zatem, jeśli $s = t$, to jedyna optymalna ścieżka (która nie zawiera operacji identycznościowych) jest pusta i ma zerowy koszt. Jeśli $s \neq t$, to z zupełności zbioru bazowych operacji edycyjnych wynika, że istnieje jedna lub więcej ścieżek edycyjnych, które przekształcają $s$ w $t$. Wszystkie te ścieżki składają się z operacji edycyjnych o ściśle dodatnim koszcie.

Niech $\gamma$ będzie kosztem ścieżki przekształcającej $s$ w $t$. Rozważmy zbiór $A$ ścieżek edycyjnych, które przekształcają $s$ w $t$ i których koszt jest ograniczony z góry przez $\gamma$. Zbiór $A$ jest niepusty i składa się z operacji edycyjnych o dodatnim koszcie mniejszym niż $\gamma$. Zbiór operacji bazowych, których koszt jest ograniczony z góry przez $\gamma$ jest skończony, co dowodzi, że zbiór $A$ jest również skończony. Ponieważ $A$ jest niepusty i skończony, to ścieżki edycyjne o mininalnym (dodatnim) koszcie istnieją i należą do $A$. Stąd, $ED(s,t) > 0$ dla $s \neq t$, tj. odległość edycyjna jest dodatnio określona.

Aby udowodnić symetrię odległości edycyjnej, rozważmy optymalną ścieżkę $E$, która przekształca $s$ w $t$, oraz odpowiadającą jej odwrotną ścieżkę $E_r$, która przekształca $t$ w $s$. Równość ich kosztów $\delta(E) = \delta(E_r)$ wynika z symetrii funkcji kosztu i symetrii zbioru operacji bazowych $\mathbb{B}$.

Aby pokazać subaddytywność, rozważmy optymalną ścieżkę $E_1$, która przekształca $s$ w $t$, optymalną ścieżkę $E_2$, która przekształca $t$ w $u$, oraz złożenie ścieżek $E_1E_2$, które przekształca $s$ w $u$. Z tego, że $\delta(E_1E_2) = \delta(E_1) + \delta(E_2) = ED(s,t) + ED(t,u)$ oraz $\delta(E_1E_2) \geq ED(s,u)$ wynika, że $ED(s,t) + ED(t,u) \geq ED(s,u)$. ■

Odległość edycyjna jest metryką, nawet gdy funkcja kosztu $\delta$ nie jest subaddytywna. Co więcej, ponieważ ciąg nakładających się operacji, które przekształcają $s$ w $t$, mogą mieć mniejszy koszt niż $\delta(s \to t)$, $\delta(s \to t)$ może być większe niż $ED(s,t)$. Rozważmy, na przykład, następujący alfbet: {`a, b, c`}, gdzie symetria i brak subaddytywności funkcji $\delta$ jest zdefiniowana

następująco:

$$\delta(\mathtt{a} \to \mathtt{c}) = \delta(\mathtt{b} \to \mathtt{c}) = 1$$
$$\delta(\mathtt{a} \to \varepsilon) = \delta(\mathtt{b} \to \varepsilon) = \delta(\mathtt{c} \to \varepsilon) = 2$$
$$\delta(\mathtt{a} \to \mathtt{b}) = 3$$

Można zobaczyć, że $3 = \delta(\mathtt{a} \to \mathtt{b}) > \delta(\mathtt{a} \to \mathtt{c}) + \delta(\mathtt{c} \to \mathtt{b}) = 2$. Stąd optymalna ścieżka edycyjna $(\mathtt{a} \to \mathtt{c}, \mathtt{c} \to \mathtt{b})$ przekształca a w b z kosztem równym 2.

**Ścisła odległość edycyjna.** Subaddytywność odległości edycyjnej pozwala używać metod właściwych przestrzeniom metrycznym. Niemniej jednak, problem minimalizacji zbioru nakładających się operacji edycyjnych, może być trudny. Aby zrównoważyć złożoność obliczeniową, zazwyczaj używana jest funkcja podobieństwa, zdefiniowana jako minimum kosztu *ścisłej ścieżki edycyjnej*. Ta ostatnia nie zawiera nakładających się na siebie operacji edycyjnych i nie modyfikuje tego samego podnapisu więcej niż raz. Odpowiająca jej odległość edycyjna nazywana jest *ścisłą odległością edycyjną* [1].

**Lemat 1.4.** *Dowolna nieścisła odległość edycyjna ogranicza z dołu odpowiadającą jej ścisłą odległość edycyjną.*

**Lemat 1.5.** *Ścisła odległość Levenshteina o jednostkowym koszcie operacji jest równa nieścisłej odległości Levenshteina o jednostkowym koszcie operacji.*

Powyższe wynika natychmiast z obserwacji, że optymalna ścieżka edycyjna zawiera jednoznakowe usunięcia, wstawienia oraz zamiany, które nigdy nie modyfikują znaku więcej niż raz.

**Lemat 1.6.** *Nieścisła odległość Damerau-Levenshteina oraz ścisła odległość Damerau-Levenshteina są różnymi funkcjami. Co więcej ścisła odległość Damerau-Levenshteina nie jest metryką, gdyż nie jest subaddytywna [1].*

*Dowód.* Ścisła odległość Damerau-Levenshteina traktuje transpozycję (tj. zamianę dwóch przylegających do siebie znaków) jako bazową operację edycyjną. Aby udowodnić lemat podamy przykład, w którym zakaz modyfikacji znaków już stransponowanych odróżnia odległość Damerau-Levenshteina od ścisłej odległości Damerau-Levenshteina [1]. ■

Rozważmy napisy ab, ba oraz acb. Z jednej strony, najkrótsza nieścisła ścieżka edycyjna, która przekształca ba w acb, tj. $(\mathtt{ba} \to \mathtt{ab}, \varepsilon \to \mathtt{c})$ zawiera dwie operacje: najpierw, zamienia znaki a i b, a następnie wstawia c pomiędzy nie. Zauważmy, że wstawienie przekształca już transformowany napis. Jednakowoż, jeśli kolejne przekształcenia tego samego podnapisu są wykluczone, to najkrótsza ścieżka edycyjna, która przekształca ba w acb, składa się z trzech operacji edycyjnych, np. $(\mathtt{ba} \to \varepsilon, \varepsilon \to \mathtt{b})$. Stąd, nieścisła odległość edycyjna jest równa dwa, podczas gdy ścisła odległość wynosi trzy [1].

Ścisła odległość Damerau-Levenshteina nie spełnia nierówności trójkąta, gdyż

$$\mathtt{ba} \xrightarrow[1]{\textit{transp. b i a}} \mathtt{ab} + \mathtt{ab} \xrightarrow[1]{\textit{wst.c}} \mathtt{acb},$$

natomiast

$$\mathtt{ba} \xrightarrow[1]{\textit{us. b}} \mathtt{a} \xrightarrow[1]{\textit{wst. c}} \mathtt{ac} \xrightarrow[1]{\textit{wst. b}} \mathtt{acb}$$

zatem

$$2 = ED(\texttt{ba}, \texttt{ab}) + ED(\texttt{ab}, \texttt{acb}) \leq ED(\texttt{ba}, \texttt{acb}) = 3.$$

Ponieważ ścisła i nieścisła odległość Damerau-Levenshteina są różnymi funkcjami, tę pierwszą nazywa się często *odległością optymalnego dopasowania napisów*. Od tego momentu w niniejszej pracy ścisłą odległość Damerau-Levenshteina nazywamy odległością optymalnego dopasowania napisów, natomiast nieścisłą odległość Damerau-Levenshteina nazywamy odległością Damerau-Levenshteina [7].

**Optymalne dopasowanie.**

Niech napisy $s$ i $t$ zostaną podzielone na tę samą liczbę, być może pustych, podnapisów: $s = s_1 s_2 \dots s_l$ i $t = t_1 t_2 \dots t_l$, takich, że $s_i \to t_i \in \mathbb{B}$. Co więcej, zakładamy, że $s_i$ i $t_j$ nie mogą być puste dla $i = j$. Mówimy, że ten podział definiuje *dopasowanie* $A = (s_1 s_2 \dots s_l, t_1 t_2 \dots t_l)$ pomiędzy napisami $s$ i $t$, w którym podnapis $s_i$ jest dopasowane do podnapisu $t_i$ [1].

Dopasowanie reprezentuje ścisła ścieżkę edycyjną $E = s_1 \to t_1, s_2 \to t_2, \dots, s_l \to t_l$. Definiujemy *koszt dopasowania* $A$ jako koszt odpowiadającej mu ścieżki edycyjnej i oznaczamy przez $\delta(A)$:

$$\delta(A) = \sum_{i=1}^{l} \delta(s_i \to t_i) \tag{1.1}$$

*Optymalne dopasowanie* to dopasowanie o najmniejszym koszcie.

PIEKNY RYSUNEK

**Przykład 1.3.** Przykład optymalnego dopasowania pomiędzy słowami `foczka` i `kozak` prezentuje rys. TUTAJNUMER. Odpowiadająca mu ścieżka edycyjna składa się z zamiany `f` $\to$ `k`, usunięcia `c` $\to \varepsilon$ oraz transpozycji `ka` $\to$ `ak`.

Warto zauważyć, że istnieje różnowartościowe (1-1) mapowanie pomiędzy zbiorem ścisłych ścieżek edycyjnych i zbiorem dopasowań: każda ścisła ścieżka edycyjna o minimalnym koszcie reprezentuje dopasowanie o najmniejszym koszcie i odwrotnie. Stąd można zastąpić problem znalezienia optymalnej ścisłej odległości edycyjnej poprzez problem znalezienia optymalnego dopasowania.

**Obliczanie odległości edycyjnej.** Główną zasadą dynamicznego algorytmu, liczącego koszt optymalnego dopasowania, jest wyrażenie kosztu dopasowania pomiędzy napisami $s$ i $t$, używając kosztu dopasowaniu ich prefiksów. Rozważmy prefiks $s_{1:i}$ o długości $i$ i prefiks $t_{1:j}$ o długości $j$ napisów $s$ i $t$ odpowiednio. Załóżmy, że $A = (s_1 s_2 \dots s_l, t_1 t_2 \dots t_l)$ jest optymalnym dopasowaniem pomiędzy $s_{1:i}$ i $t_{1:j}$, którego koszt oznaczamy przez $C_{i,j}$ [1].

Używając równania 1.1 oraz definicji optymalnego dopasowania, łatwo pokazać, że $C_{i,j}$ może zostać policzone przy użyciu następującej ogólnej rekurencji [6]:

$$C_{0,0} = 0 \tag{1.2}$$

$$C_{i,j} = min\{\delta(s_{i':i} \to t_{j':j}) + C_{i'-1,j'-1} | s_{i':i} \to t_{j':j} \in \mathbb{B}\} \tag{1.3}$$

Można zauważyć, że:

- koszt dopasowania napisów $s$ i $t$ jest równy $C_{|s|,|t|}$;
- wszystkie optymalne dopasowania mogą zostać wyznaczone poprzez odwracanie rekurencji 1.2 (przechodzenie od tyłu).

Rozważmy teraz odległość Hamminga, gdzie $s_{i':i} \to t_{j':j}$ to zamiany znaków o koszcie równym jeden. Stąd,

$$\delta(s_{i':i} \to t_{j':j}) = [s_{i':i} \neq t_{j':j}] \tag{1.4}$$

gdzie $[X]$ jest równe jeden, gdy warunek $X$ jest spełniony, zero w przeciwnym przypadku. Co więcej, w tym przypadku możliwa jest tylko jedna kombinacja $i'$ oraz $j'$, mianowicie $i' = i$ oraz $j' = j$. Co więcej, odległość ta jest zdefiniowana jedynie dla $|s| = |t|$, zatem $C_{i,j}$ może być policzone jedynie dla $i = j$. Wówczas definicja odległości Hamminga nie jest rekursywna (rekursyjna?) i można ją zapisać następująco:

**Definicja 1.3.** Odległością Hamminga *nazywamy [2]:*

$$d_{hamming}(s,t) = \begin{cases} \sum_{i=1}^{|s|} \delta(s_i \to t_i) = \sum_{i=1}^{|s|} [s_i \neq t_i], & gdy\ |s| = |t|, \\ \infty, & w\ przeciwnym\ przypadku, \end{cases}$$

Intuicyjnie rzecz biorąc odległość Hamminga zlicza liczbę indeksów, na których dwa napisy mają różny znak. Odległość ta przyjmuje wartości ze zbioru $\{0, \ldots, |s|\}$, gdy $|s| = |t|$, natomiast jest równa nieskończoności, gdy napisy mają różne długości.

[PIĘKNY RYSUNEK??]

**Przykład 1.4.** Odległość Hamminga między słowami `koza` i `foka` wynosi $d_{hamming}(\texttt{koza}, \texttt{foka}) = 2$, natomiast między słowami `koza` i `foczka` wynosi ona $d_{hamming}(\texttt{koza}, \texttt{foczka}) = \infty$.

Rozważmy teraz odległość najdłuższego wspólnego podnapisu, gdzie $s_{i':i} \to t_{j':j}$ to wstawienia i usunięcia znaków o koszcie równym jeden. Wówczas istnieją dwie kombinacje $i'$ oraz $j'$ z równania 1.4, odpowiadające usunięciu i wstawieniu, odpowiednio:

- $i' = i - 1$ oraz $j' = j$,
- $i' = i$ oraz $j' = j - 1$.

Uwzględniając powyższe uproszczenia, możemy następująco przepisać ogólną postać rekurencji 1.2 dla odległości najdłuższego wspólnego podnapisu:

$$C_{i,j} = min \begin{cases} 0, & \text{gdy } i = j = 0 \\ C_{i-1,j} + 1, & \text{gdy } i > 0 \\ C_{i,j-1} + 1, & \text{gdy } j > 0 \end{cases} \tag{1.5}$$

Odległość najdłuższego wspólnego przyjmuje wartości ze zbioru $\{0, |s| + |t|\}$, przy czym maksimum jest osiągane, gdy $s$ i $t$ nie mają ani jednego wspólnego znaku. Odległość tę oznaczamy przez $d_{lcs}$.

**Przykład 1.5.** Odległość najdłuższego wspólnego podnapisu między napisami `koza` i `foka` wynosi: $d_{lsc}(\texttt{koza}, \texttt{foka}) = 4$, bo $\texttt{koza} \xrightarrow[1]{us.\ k} \texttt{oza} \xrightarrow[1]{us.\ z} \texttt{oa} \xrightarrow[1]{wst.\ f} \texttt{foa} \xrightarrow[1]{wst.\ k} \texttt{foka}$.

Powyższy przykład pokazuje, że w ogólności nie ma unikalnej najkrótszej drogi transformacji jednego napisu w drugi, gdyż można zamienić kolejność usuwania (lub wstawiania) znaków i również uzyskać odległość równą 4.

Jak sugeruje nazwa, odległość najdłuższego wspólnego podnapisu, ma też inną interpretację. Poprzez wyrażenie *najdłuższy wspólny podnapis* rozumiemy najdłuższy ciąg utworzony przez sparowanie znaków z $s$ i $t$ nie zmieniając ich porządku. Wówczas odległość ta jest rozumiana jako liczba niesparowanych znaków z obu napisów. W powyższym przykładzie może to być zwizualizowane następująco:

[PIĘKNY RYSUNEK]

Jak widać na rysunku, znaki `k`, `z`, `f` i `k` pozostają bez pary, dając odległość równą 4.

Przejdźmy do odległości Levenshteina. Odległość ta dopuszcza, oprócz usunięć i wstawień, zamiany znaków, zatem istnieją trzy kombinacje $i'$ oraz $j'$ z równania 1.4:

- $i' = i - 1$ oraz $j' = j$,
- $i' = i$ oraz $j' = j - 1$,
- $i' = i - 1$ oraz $j' = j - 1$.

Stąd ogólna postać rekurencji 1.2 dla odległości Levenshteina może zostać przepisana następująco:

$$C_{i,j} = min \begin{cases} 0, & \text{gdy } i = j = 0 \\ C_{i-1,j} + 1, & \text{gdy } i > 0 \\ C_{i,j-1} + 1, & \text{gdy } j > 0 \\ C_{i-1,j-1} + [s_i \neq t_j], & \text{gdy } i, j > 0 \end{cases} \tag{1.6}$$

Odległość Levenshteina oznaczamy przez $d_{lv}$.

**Przykład 1.6.** Odległość Levenshteina między napisami `koza` i `foka` wynosi: $d_{lv}(\texttt{koza}, \texttt{foka}) = 2$, bo $\texttt{koza} \xrightarrow[1]{zm.\ k\ na\ f} \texttt{foza} \xrightarrow[1]{zm.\ z\ na\ k} \texttt{foka}$.

Powyższy przykład ilustruje, że dodatkowa elastyczność w porównaniu do odległości najdłuższego wspólnego podnapisu, daje mniejszą wartość odległości między napisami, jako że potrzebujemy jedynie dwóch zamian znaków [7].

Przypomnijmy, że mówimy o uogólnionej odległości, gdy zmienimy koszty poszczególnych operacji na różne od jeden. Gdy za koszt przyjmiemy np. $(0.1, 1, 0.3)$ dla usunięć, wstawień i zamian znaków odpowiednio, to uogólniona odległość Levenshteina między napisami `koza` i `foka` wynosi: $d_{lv}(\texttt{koza}, \texttt{foka}) = 0.6$, bo $\texttt{koza} \xrightarrow[0.3]{zm.\ k\ na\ f} \texttt{foza} \xrightarrow[0.3]{zm.\ z\ na\ k} \texttt{foka}$.

Uogólniona odległość Levenshteina spełnia definicję metryki, gdy koszt usunięcia jest równy kosztowi wstawienia znaku. W przeciwnym przypadku nie spełnia ona założenia o symetrii. Jednakowoż, symetria zostaje zachowana przy jednoczesnej zamianie $s$ i $t$ oraz kosztów usunięcia i wstawienia znaku, jako że liczba usunięć znaków przy przetwarzaniu napisu $s$ w napis $t$ jest równa liczbie wstawień znaków przy przetwarzaniu napisu $t$ w napis $s$ [7]. Dobrze obrazuje to następujący przykład.

**Przykład 1.7.** Przyjmijmy za koszt usunięcia, wstawienia i zamiany znaku $(0.1, 1, 0.3)$ odpowiednio. Wówczas uogólniona odległość Levenshteina dla napisów `koza` i `foczka` wynosi:

$$d_{lv}(\texttt{koza}, \texttt{foczka}) = 0.5, \tag{1.7}$$

bo

$$\texttt{koza} \xrightarrow[0.3]{zm.\ k\ na\ f} \texttt{foza} \xrightarrow[0.1]{wst.c} \texttt{focza} \xrightarrow[0.1]{wst.k} \texttt{foczka},$$

natomiast

$$d_{lv}(\texttt{foczka}, \texttt{koza}) = 2.3, \tag{1.8}$$

bo

$$\texttt{foczka} \xrightarrow[0.3]{zm.\ f\ na\ k} \texttt{koczka} \xrightarrow[1]{us.c} \texttt{kozka} \xrightarrow[1]{us.k} \texttt{koza}.$$

Gdy za koszty przyjmiemy $(1, 0.1, 0.3)$, to uogólniona odległość Levenshteina wynosi:

$$d_{lv}(\texttt{koza}, \texttt{foczka}) = 2.3,$$

bo

$$\texttt{koza} \xrightarrow[0.3]{zm.\ k\ na\ f} \texttt{foza} \xrightarrow[1]{wst.c} \texttt{focza} \xrightarrow[1]{wst.\ k} \texttt{foczka},$$

czyli analogicznie, jak w przypadku 1.8. Natomiast

$$d_{lv}(\texttt{foczka}, \texttt{koza}) = 0.5,$$

bo

$$\texttt{foczka} \xrightarrow[0.3]{zm.\ f\ na\ k} \texttt{koczka} \xrightarrow[0.1]{us.c|} kozka \xrightarrow[0.1]{us.\ k} \texttt{koza},$$

czyli analogicznie, jak w przypadku 1.7.

Zgodnie z lematem 1.5 nieścisła odległość Levenshteina jest równa ścisłej odległości Levenshteina. Z drugiej strony, ścisła odległość edycyjna jest równa kosztowi optymalnego dopasowania. Stąd rekurencja 1.2 liczy nieścisłą odległość Levenshteina.

**Definicja 1.4.** Odległością optymalnego dopasowania napisów *na* $\Sigma^*$ *nazywamy:*

$$d_{osa}(s,t) = \begin{cases} 0, & gdy\ s = t = \varepsilon, \\ min\{ & \\ \quad d_{osa}(s, t_{1:|t|-1}) + w_1, & \\ \quad d_{osa}(s_{1:|s|-1}, t) + w_2, & \\ \quad d_{osa}(s_{1:|s|-1}, t_{1:|t|-1}) + [1 - \delta(s_{|s|}, t_{|t|})]w_3 & \\ \quad d_{osa}(s_{1:|s|-2}, t_{1:|t|-2}) + w_4,\ gdy\ s_{|s|} = t_{|t|-1}, s_{|s|-1} = t_{|t|} & \\ \quad \}, & w\ przeciwnym\ przypadku, \end{cases}$$

*gdzie* $w_1, w_2$, $w_3$ *iw*$_4$ *to niezerowe liczby rzeczywiste, oznaczające kary za odpowiednio usunięcie, wstawienie, zamianę oraz transpozycję znaków.*

Odległość optymalnego dopasowania napisów jest bezpośrednim rozszerzeniem odległości Levenshteina, która zlicza również liczbę transpozycji przylegających znaków, potrzebnych do przetworzenia jednego napisu w drugi. W przeciwieństwie do wcześniej zaprezentowanych odległości, nie spełnia ona nierówności trójkąta, tj. podpunktu **??** z definicji **??** [7]:

$$2 = d_{osa}(\texttt{ba}, \texttt{ab}) + d_{osa}(\texttt{ab}, \texttt{acb}) \leq d_{osa}(\texttt{ba}, \texttt{acb}) = 3,$$

gdyż

$$\texttt{ba} \xrightarrow[1]{\textit{transp. b i a}} \texttt{ab} + \texttt{ab} \xrightarrow[1]{\textit{wst.c}} \texttt{acb},$$

natomiast

$$\texttt{ba} \xrightarrow[1]{\textit{us. b}} \texttt{a} \xrightarrow[1]{\textit{wst. c}} \texttt{ac} \xrightarrow[1]{\textit{wst. b}} \texttt{acb}.$$

W ostatnim przykładzie, zmniejszenie odległości poprzez zamianę liter $|a|$ i $|b|$, a następnie wstwienie litery $|c|$ spowodowałoby dwukrotne przekształcenie tego samego podnapisu. Z tego powodu odległość optymalnego dopasowania napisów bywa również nazywana *ścisłą odległością Damerau-Levenshteina* i jest często mylona z właściwą *odległością Damerau-Levenshteina*. Ta ostatnia pozwala na przekształcanie tego samego podnapisu wielokrotnie i jest metryką w rozumieniu definicjij **??**, ale nie spełnia założenia o nie przekształcaniu wielokrotnie tego samego podnapisu [7].

[MIARA DAMERAU-LEVENSHTEINA]

W przypadku odległości Levenshteina i odległości optymalnego dopasowania napisów, maksymalna odległość między napisami $s$ i $t$ wynosi $max\{|s|, |t|\}$. Jednakowoż, gdy liczba dopuszczalnych operacji edycyjnych rośnie, to liczba dopuszczalnych ścieżek między napisami wzrasta, co pozwala ewentualnie zmiejszyć odległość między napisami. Dlatego relację między zaprezentowanymi powyżej odległościami można podsumować następująco [7]:

$$\left.\begin{array}{r} \infty \geq |s| \geq d_{hamming}(s,t) \\ |s| + |t| \geq d_{lcs}(s,t) \\ max\{|s|, |t|\} \end{array}\right\} \geq d_{lv}(s,t) \geq d_{osa}(s,t) \geq 0.$$

# Literatura


[1] Leonid Boytsov. Indexing methods for approximate dictionary searching: Comparative analysis. *J. Exp. Algorithmics*, 16:1.1:1.1–1.1:1.91, Maj 2011.

[2] R. W. Hamming. Error Detecting and Error Correcting Codes. *Bell System Techincal Journal*, 29:147–160, 1950.

[3] V. Levenshtein. Binary codes capable of correcting spurious insertions and deletions of ones. *Problems of Information Transmission*, 1:8–17, 1965.

[4] G. Navarro. A guided tour to approximate string matching. *ACM Computing Surveys*, 33(1):31–88, 2001.

[5] Saul B. Needleman, Christian D. Wunsch. A general method applicable to the search for similarities in the amino acid sequence of two proteins. *Journal of Molecular Biology*, 48(3):443–453, Marzec 1970.

[6] Esko Ukkonen. Algorithms for approximate string matching. *Inf. Control*, 64(1-3):100–118, Marzec 1985.

[7] Mark P. J. van der Loo. The stringdist Package for Approximate String Matching. *The R Journal*, 6:111–122, 2014.

[8] Robert A. Wagner, Michael J. Fischer. The string-to-string correction problem. *Journal of the ACM*, 21(1):168–173, January 1974.

[9] Robert A. Wagner, Roy Lowrance. An extension of the string-to-string correction problem. *J. ACM*, 22(2):177–183, Kwiecie/n 1975.