Chapter 11
Distances on Strings and Permutations

✓ An *alphabet* is a finite set $\mathcal{A}$, $|\mathcal{A}| \geq 2$, elements of which are called *characters* (or *symbols*). A *string* (or *word*) is a sequence of characters over a given finite alphabet $\mathcal{A}$. The set of all finite strings over the alphabet $\mathcal{A}$ is denoted by $W(\mathcal{A})$. Examples of real world applications, using distances and similarities of string pairs, are Speech Recognition, Bioinformatics, Information Retrieval, Machine Translation, Lexicography, Dialectology.

✓ A *substring* (or *factor*, *chain*, *block*) of the string $x = x_1 \dots x_n$ is any contiguous subsequence $x_i x_{i+1} \dots x_k$ with $1 \leq i \leq k \leq n$. A *prefix* of a string $x_1 \dots x_n$ is any substring of it starting with x_1; a *suffix* is any substring of it finishing with x_n. If a string is a part of a text, then the *delimiters* (a space, a dot, a comma, etc.) are added to the alphabet $\mathcal{A}$.

✓ A *vector* is any finite sequence consisting of real numbers, i.e., a finite string over the *infinite alphabet* $\mathbb{R}$. A *frequency vector* (or *discrete probability distribution*) is any string $x_1 \dots x_n$ with all $x_i \geq 0$ and $\sum_{i=1}^{n} x_i = 1$. A *permutation* (or *ranking*) is any string $x_1 \dots x_n$ with all x_i being different numbers from $\{1, \dots, n\}$.

✓ An *editing operation* is an operation on strings, i.e., a *symmetric binary relation* on the set of all considered strings. Given a set of editing operations $\mathcal{O} = \{O_1, \dots, O_m\}$, the corresponding **editing metric** (or *unit cost edit distance*) between strings x and y is the minimum number of editing operations from $\mathcal{O}$ needed to obtain y from x. It is the **path metric** of a graph with the vertex-set $W(\mathcal{A})$ and xy being an edge if y can be obtained from x by one of the operations from $\mathcal{O}$. In some applications, a *cost function* is assigned to each type of editing operation; then the editing distance is the minimal total cost of transforming x into y. Given a set of editing operations $\mathcal{O}$ on strings, the corresponding **necklace editing metric** between cyclic strings x and y is the minimum number of editing operations from $\mathcal{O}$ needed to obtain y from x, minimized over all rotations of x.

✓ The main editing operations on strings are:

- *Character indel*, i.e., insertion or deletion of a character
- *Character replacement*
- *Character swap*, i.e., an interchange of adjacent characters

M.M. Deza and E. Deza, *Encyclopedia of Distances*,
DOI 10.1007/978-3-642-00234-2_11, © Springer-Verlag Berlin Heidelberg 2009

- *Substring move*, i.e., transforming, say, the string $x = x_1 \dots x_n$ into the string $x_1 \dots x_{i-1}\mathbf{x_j} \dots \mathbf{x_{k-1}}x_i \dots x_{j-1}x_k \dots x_n$
- *Substring copy*, i.e., transforming, say, $x = x_1 \dots x_n$ into $x_1 \dots x_{i-1}\mathbf{x_j} \dots \mathbf{x_{k-1}}x_i \dots x_n$
- *Substring uncopy*, i.e., the removal of a substring provided that a copy of it remains in the string

We list below the main distances on strings. However, some string distances will appear in Chaps. 15, 21 and 23, where they fit better, with respect to the needed level of generalization or specification.

11.1 Distances on general strings

- ✓ **Levenstein metric**
  The **Levenstein metric** (or **edit distance**, *shuffle-Hamming distance*, *Hamming+Gap metric*) is (Levenstein 1965) an editing metric on $W(\mathcal{A})$, obtained for $\mathcal{O}$ consisting of only character replacements and indels.

  The Levenstein metric $d_L(x, y)$ between strings $x = x_1 \dots x_m$ and $y = y_1 \dots y_n$ is equal to
  $$\min\{d_H(x^*, y^*)\},$$
  where x^*, y^* are strings of length k, $k \geq \max\{m, n\}$, over the alphabet $\mathcal{A}^* = \mathcal{A} \cup \{*\}$ so that, after deleting all new characters $*$, strings x^* and y^* shrink to x and y, respectively. Here, the *gap* is the new symbol $*$, and x^*, y^* are *shuffles* of strings x and y with strings consisting of only $*$.

  The *Levenstein similarity* is $1 - \frac{d_L(x,y)}{\max\{m,n\}}$.

  The **Damerau–Levenstein metric** (Damerau 1964) is an editing metric on $W(\mathcal{A})$, obtained for $\mathcal{O}$ consisting only of character replacements, indels and transpositions. In the Levenstein metric, a transposition corresponds to two editing operations: one insertion and one deletion.

  The **constrained edit distance** (Oomen 1986) is the Levenstein metric, but the ranges for the number of replacements, insertions and deletions are specified.
- **Editing metric with moves**
  The **editing metric with moves** is an editing metric on $W(\mathcal{A})$ [Corm03], obtained for $\mathcal{O}$ consisting of only substring moves and indels.
- **Editing compression metric**
  The **editing compression metric** is an editing metric on $W(\mathcal{A})$ [Corm03], obtained for $\mathcal{O}$ consisting of only indels, copy and uncopy operations.
- **Indel metric**
  The **indel metric** is an editing metric on $W(\mathcal{A})$, obtained for $\mathcal{O}$ consisting of only indels.

It is an analog of the **Hamming metric** $|X \Delta Y|$ between sets X and Y. For strings $x = x_1 \dots x_m$ and $y = y_1 \dots y_n$ it is equal to $m + n - 2LCS(x, y)$, where the similarity $LCS(x, y)$ is the length of the longest common subsequence of x and y.

The **factor distance** on $W(\mathcal{A})$ is $m+n-2LCF(x,y)$, where the similarity $LCF(x,y)$ is the length of the longest common substring (factor) of x and y.

The *LCS ratio* and the *LCF ratio* are the similarities on $W(\mathcal{A})$ defined by $\frac{LCS(x,y)}{\min\{m,n\}}$ and $\frac{LCF(x,y)}{\min\{m,n\}}$, respectively; sometimes, the denominator is $\max\{m,n\}$ or $\frac{m+n}{2}$.

- ✓ **Swap metric**

  The **swap metric** is an editing metric on $W(\mathcal{A})$, obtained for $\mathcal{O}$ consisting only of character swaps.

- ✓ **Edit distance with costs**

  Given a set of editing operations $\mathcal{O} = \{O_1, \dots, O_m\}$ and a *weight* (or *cost function*) $w_i \geq 0$, assigned to each type O_i of operation, the **edit distance with costs** between strings x and y is the minimal total cost of an *editing path* between them, i.e., the minimal sum of weights for a sequence of operations transforming x into y.

  The **normalized edit distance** between strings x and y (Marzal and Vidal 1993) is the minimum, over all editing paths P between them, of $\frac{W(P)}{L(P)}$, where $W(P)$ and $L(P)$ are the total cost and the length of the editing path P.

- **Transduction edit distances**

  The **Levenstein metric** with costs between strings x and y is modeled in [RiYi98] as a memoryless stochastic transduction between x and y.

  Each step of transduction generates either a character replacement pair (a, b), a deletion pair $(a, \emptyset)$, an insertion pair $(\emptyset, b)$, or the specific termination symbol t according to a probability function $\delta : E \cup \{t\} \to [0, 1]$, where E is the set of all possible above pairs. Such a transducer induces a probability function on the set of all sequences of operations.

  The **transduction edit distances** between strings x and y are [RiYi98] $\ln p$ of the following probabilities p:

  for the **Viterbi edit distance**, the probability p of the most likely sequence of editing operations transforming x into y;

  for the **stochastic edit distance**, the probability p of the string pair (x, y).

  Those distances are never zero unless they are infinite for all other string pairs.

  This model allows one to learn (in order to reduce error rate) the edit costs for the Levenstein metric from a corpus of examples (training set of string pairs). This learning is automatic; it reduces to estimating the parameters of above transducer.

- ✓ **Bag distance**
  The **bag distance** (or *multiset metric*, *counting filter*) is a metric on $W(\mathcal{A})$, defined (Navarro 1997) by

  $$\max\{|X \backslash Y|, |Y \backslash X|\}$$

  for any strings x and y, where X and Y are the *bags of symbols* (multisets of characters) in strings x and y, respectively, and, say, $|X \backslash Y|$ counts the number of elements in the multiset $X \backslash Y$. Cf. **metrics between multisets** in Chap. 1.

  The bag distance is a (computationally) cheap approximation of the **Levenstein metric**.
- **Marking metric**
  The **marking metric** is a metric on $W(\mathcal{A})$ [EhHa88], defined by

  $$\log_2((diff(x,y)+1)(diff(y,x)+1))$$

  for any strings $x = x_1 \dots x_m$ and $y = y_1 \dots y_n$, where $diff(x,y)$ is the minimal size $|M|$ of a subset $M \subset \{1, \dots, m\}$ such that any substring of x, not containing any x_i with $i \in M$, is a substring of y.

  Another metric, defined in [EhHa88], is $\log_2(diff(x,y)+diff(y,x)+1)$.
- ✓ **Transformation distance**
  The **transformation distance** is an **editing distance with costs** on $W(\mathcal{A})$ (Varre, Delahaye and Rivals 1999) obtained for $\mathcal{O}$ consisting only of substring copy, uncopy and substring indels. The distance between strings x and y is the minimal cost of transformation x into y using these operations, where the cost of each operation is the length of its description. For example, the description of the copy requires a binary code specifying the type of operation, an offset between the substring locations in x and in y, and the length of the substring. A code for insertion specifies the type of operation, the length of the substring and the sequence of the substring.
- **L_1 rearrangement distance**
  The **L_1 rearrangement distance** (Amir, Aumann, Indyk, Levy and Porat 2007) between strings $x = x_1 \dots x_m$ and $y = y_1 \dots y_m$ is equal to

  $$\min_{\pi} \sum_{i=1}^{m} |i - \pi(i)|,$$

  where $\pi : \{1, \dots, m\} \to \{1, \dots, m\}$ is a permutation transforming x into y; if there are no such permutations, the distance is equal to ∞.

  The **L_∞ rearrangement distance** (Amir, Aumann, Indyk, Levy and Porat 2007) between strings $x = x_1 \dots x_m$ and $y = y_1 \dots y_m$ is $\min_{\pi} \max_{1 \le i \le m} |i - \pi(i)|$ and, again, it is ∞ if such a permutation does not exist.

  Cf. **genome rearrangement distances** in Chap. 23.

- **Normalized information distance**
The **normalized information distance** d between two binary strings x and y is a symmetric function on $W(\{0,1\})$ [LCLM04], defined by

$$\frac{\max\{K(x|y^*), K(y|x^*)\}}{\max\{K(x), K(y)\}}$$

Here, for binary strings u and v, u^* is a shortest binary program to compute u on an appropriate (i.e., using a *Turing-complete* language) universal computer, the *Kolmogorov complexity* (or *algorithmic entropy*) $K(u)$ is the length of u^* (the ultimate compressed version of u), and $K(u|v)$ is the length of the shortest program to compute u if v is provided as an auxiliary input.

The function $d(x,y)$ is a metric up to small error term: $d(x,x) = O((K(x))^{-1})$, and $d(x,z) - d(x,y) - d(y,z) = O((\max\{K(x), K(y), K(z)\})^{-1})$. (Cf. $d(x,y)$ the **information metric** (or *entropy metric*) $H(X|Y) + H(Y|X)$ between stochastic sources X and Y.)

The Kolmogorov complexity is uncomputable and depends on the chosen computer language; so, instead of $K(u)$, were proposed the *minimum message length* (shortest overall message) by Wallace (1968) and the *minimum description length* (largest compression of data) by Rissanen (1978).

The **normalized compression distance** is a distance on $W(\{0,1\})$ [LCLM04], [BGLVZ98], defined by

$$\frac{C(xy) - \min\{C(x), C(y)\}}{\max\{C(x), C(y)\}}$$

for any binary strings x and y, where $C(x), C(y)$, and $C(xy)$ denote the size of the compression (by fixed compressor C, such as gzip, bzip2, or PPMZ) of strings x, y, and their *concatenation* xy. This distance is not a metric. It is an approximation of the normalized information distance. A similar distance is defined by $\frac{C(xy)}{C(x)+C(y)} - \frac{1}{2}$.

- **Lempel–Ziv distance**
The **Lempel–Ziv distance** between two binary strings x and y of length n is

$$\max\{\frac{LZ(x|y)}{LZ(x)}, \frac{LZ(y|x)}{LZ(y)}\},$$

where $LZ(x) = \frac{|P(x)| \log |P(x)|}{n}$ is the *Lempel–Ziv complexity* of x, approximating its *Kolmogorov complexity* $K(x)$. Here $P(x)$ is the set of non-overlapping substrings into which x is parsed sequentially, so that the new substring is not yet contained in the set of substrings generated so far. For example, such a *Lempel–Ziv parsing* for $x = 001100101010011$ is $0|01|1|00|10|101|001|11$. Now, $LZ(x|y) = \frac{|P(x) \backslash P(y)| \log |P(x) \backslash P(y)|}{n}$.

- **Anthony–Hammer similarity**
  The **Anthony–Hammer similarity** between a binary string $x = x_1 \dots x_n$ and the set Y of binary strings $y = y_1 \dots y_n$ is the maximal number m such that, for every m-subset $M \subset \{1, \dots, n\}$, the substring of x, containing only x_i with $i \in M$, is a substring of some $y \in Y$ containing only y_i with $i \in M$.
- ✓ **Jaro similarity**
  Given strings $x = x_1 \dots x_m$ and $y = y_1 \dots y_n$, call a character x_i *common with* y if $x_i = y_j$, where $|i - j| \leq \frac{\min\{m,n\}}{2}$. Let $x' = x'_1 \dots x'_{m'}$ be all the characters of x which are common with y (in the same order as they appear in x), and let $y' = y'_1 \dots y'_{n'}$ be the analogous string for y.

  The **Jaro similarity** $Jaro(x, y)$ between strings x and y is defined by

$$\frac{1}{3}\left(\frac{m'}{m} + \frac{n'}{n} + \frac{|\{1 \leq i \leq \min\{m', n'\} : x'_i = y'_i\}|}{\min\{m', n'\}}\right).$$

  This and following two similarities are used in Record Linkage.
- **Jaro–Winkler similarity**
  The **Jaro–Winkler similarity** between strings x and y is defined by

$$Jaro(x, y) + \frac{\max\{4, LCP(x, y)\}}{10}(1 - Jaro(x, y)),$$

  where $Jaro(x, y)$ is the **Jaro similarity**, and $LCP(x, y)$ is the length of the longest common prefix of x and y.
- ✓ **q-gram similarity**
  Given an integer $q \geq 1$ (usually, q is 2 or 3), the **q-gram similarity** between strings x and y is defined by

$$\frac{2q(x, y)}{q(x) + q(y)},$$

  where $q(x)$, $q(y)$ and $q(x, y)$ are the sizes of multisets of all *q-grams* (substrings of length q) occurring in x, y and both of them, respectively. Sometimes, $q(x, y)$ is divided not by the average of $q(x)$ and $q(y)$, as above, but by their minimum, maximum or *harmonic mean* $\frac{2q(x)q(y)}{q(x)+q(y)}$. Cf. **metrics between multisets** in Chap. 1 and, in Chap. 17, **Dice similarity**, **Simpson similarity**, **Braun–Blanquet similarity** and **Anderberg similarity**.

  Sometimes, the strings x and y are *padded* before computing their q-gram similarity, i.e., $q - 1$ special characters are added to their beginnings and ends. Padding increases the matching quality since q-grams at the beginning and end of strings are q-grams not matched to other q-grams.

  The q-gram similarity is an example of **token-based similarities**, i.e., ones defined in terms of *tokens* (selected substrings or words). Here tokens are q-grams. A generic **dictionary-based metric** between strings x and y is $|D(x)\Delta D(y)|$, where $D(z)$ denotes the full *dictionary* of z, i.e., the set of all of its substrings.

- **Prefix-Hamming metric**
  The **prefix-Hamming metric** between strings $x = x_1 \dots x_m$ and $y = y_1 \dots y_n$ is defined by

  $$(\max\{m,n\} - \min\{m,n\}) + |\{1 \le i \le \min\{m,n\} : x_i \ne y_i\}|.$$

- **Weighted Hamming metric**
  If $(\mathcal{A}, d)$ is a metric space, then the **weighted Hamming metric** between strings $x = x_1 \dots x_m$ and $y = y_1 \dots y_m$ is defined by

  $$\sum_{i=1}^{m} d(x_i, y_i).$$

  The term *weighted Hamming metric* (or *weighted Hamming distance*) is also used for $\sum_{1 \le i \le m, x_i \ne y_i} w_i$, where, for any $1 \le i \le m$, $w(i) > 0$ is its *weight*.
- **Fuzzy Hamming distance**
  If $(\mathcal{A}, d)$ is a metric space, the **fuzzy Hamming distance** between strings $x = x_1 \dots x_m$ and $y = y_1 \dots y_m$ is an **editing distance with costs** on $W(\mathcal{A})$ obtained for $\mathcal{O}$ consisting of only indels, each of fixed cost $q > 0$, and *character shifts* (i.e., moves of 1-character substrings), where the cost of replacement of i by j is a function $f(|i-j|)$. This distance is the minimal total cost of transforming x into y by these operations. Bookstein, Klein, Raita (2001) introduced this distance for Information Retrieval and proved that it is a metric if f is a monotonically increasing concave function on integers vanishing only at 0. The case $f(|i-j|) = C|i-j|$, where $C > 0$ is a constant and $|i-j|$ is a time shift, corresponds to the Victor–Purpura **spike train distance** in Chap. 23.

  Ralescu (2003) introduced, for Image Retrieval, another **fuzzy Hamming distance** on $\mathcal{R}^m$. The **Ralescu distance** between two strings $x = x_1 \dots x_m$ and $y = y_1 \dots y_m$ is the fuzzy cardinality of the difference fuzzy set $D_\alpha(x,y)$ (where α is a parameter) with membership function

  $$\mu_i = 1 - e^{-\alpha(x_i - y_i)^2}, 1 \le i \le m.$$

  The *non-fuzzy cardinality of the fuzzy set* $D_\alpha(x,y)$ approximating its fuzzy cardinality is $|\{1 \le i \le m : \mu_i > \frac{1}{2}\}|$.
- **Needleman–Wunsch–Sellers metric**
  If $(\mathcal{A}, d)$ is a metric space, the **Needleman–Wunsch–Sellers metric** (or *global alignment metric*) is an **editing distance with costs** on $W(\mathcal{A})$ [NeWu70], obtained for $\mathcal{O}$ consisting of only indels, each of fixed

cost $q > 0$, and character replacements, where the cost of replacement of i by j is $d(i, j)$. This metric is the minimal total cost of transforming x into y by these operations. Equivalently, it is

$$\min\{d_{wH}(x^*, y^*)\},$$

where x^*, y^* are strings of length k, $k \geq \max\{m, n\}$, over the alphabet $\mathcal{A}^* = \mathcal{A} \cup \{*\}$, so that, after deleting all new characters $*$, strings x^* and y^* shrink to x and y, respectively. Here $d_{wH}(x^*, y^*)$ is the **weighted Hamming metric** between x^* and y^* with weight $d(x_i^*, y_i^*) = q$ (i.e., the editing operation is an indel) if one of x_i^*, y_i^* is $*$, and $d(x_i^*, y_i^*) = d(i, j)$, otherwise.

The **Gotoh–Smith–Waterman distance** (or *string distance with affine gaps*) is a more specialized editing metric with costs (see [Goto82]). It discounts mismatching parts at the beginning and end of the strings x, y, and introduces two indel costs: one for starting an *affine gap* (contiguous block of indels), and another one (lower) for extending a gap.

- **Duncan metric**

  Consider the set X of all strictly increasing infinite sequences $x = \{x_n\}_n$ of positive integers. Define $N(n, x)$ as the number of elements in $x = \{x_n\}_n$ which are less than n, and $\delta(x)$ as the *density* of x, i.e., $\delta(x) = \lim_{n\to\infty} \frac{N(n,x)}{n}$. Let Y be the subset of X consisting of all sequences $x = \{x_n\}_n$ for which $\delta(x) < \infty$.

  The **Duncan metric** is a metric on Y, defined, for $x \neq y$, by

$$\frac{1}{1 + LCP(x, y)} + |\delta(x) - \delta(y)|,$$

  where $LCP(x, y)$ is the length of the longest common prefix of x and y.

- **Martin metric**

  The **Martin metric** d^a between strings $x = x_1 \dots x_m$ and $y = y_1 \dots y_n$ is defined by

$$|2^{-m} - 2^{-n}| + \sum_{t=1}^{\max\{m,n\}} \frac{a_t}{|\mathcal{A}|^t} \sup_z |k(z, x) - k(z, y)|,$$

  where z is any string of length t, $k(z, x)$ is the *Martin kernel* of a *Markov chain* $M = \{M_t\}_{t=0}^{\infty}$, and the sequence $a \in \{a = \{a_t\}_{t=0}^{\infty} : a_t > 0, \sum_{t=1}^{\infty} a_t < \infty\}$ is a parameter.

- **Baire metric**

  The **Baire metric** is an ultrametric between finite or infinite strings x and y, defined, for $x \neq y$, by

$$\frac{1}{1 + LCP(x, y)},$$

where $LCP(x, y)$ is the length of the longest common prefix of x and y. Cf. **Baire space** in Chap. 2.

Given an infinite *cardinal number* κ and a set A of cardinality κ, the Cartesian product of countably many copies of A endowed with above ultametric $\frac{1}{1+LCP(x,y)}$ is called the **Baire space of weight** κ and denoted by $B(\kappa)$. In particular, $B(\aleph_0)$ (called the *Baire zero-dimensional space*) is homeomorphic to the space Irr of irrationals with **continued fraction metric** (cf. Chap. 12).

- **Generalized Cantor metric**
  The **generalized Cantor metric** (or, sometimes, *Baire distance*) is an ultrametric between infinite strings x and y, defined, for $x \neq y$, by

$$a^{1+LCP(x,y)},$$

  where a is a fixed number from the interval $(0, 1)$, and $LCP(x, y)$ is the length of the longest common prefix of x and y.

  This ultrametric space is **compact**. In the case $a = \frac{1}{2}$, the metric $\frac{1}{2^{1+LCP(x,y)}}$ was considered on a remarkable **fractal** (cf. Chap. 1) from $[0, 1]$, the *Cantor set*; cf. **Cantor metric** in Chap. 18.

  Comyn and Dauchet (1985) and Kwiatkowska (1990) introduced some analogues of generalized Cantor metric for *traces*, i.e., equivalence classes of strings with respect to a congruence relation identifying strings x, y that are identical up to permutation of concurrent actions ($xy = yx$).

- **Parentheses string metrics**
  Let P_n be the set of all strings on the alphabet $\{(,)\}$ generated by a grammar and having n open and n closed parentheses. A **parentheses string metric** is an editing metric on P_n (or on its subset) corresponding to a given set of editing operations.

  For example, the **Monjardet metric** (Monjardet 1981) between two parentheses strings $x, y \in P_n$ is the minimum number of adjacent parentheses interchanges [“()” to “)(” or “)(” to “()”] needed to obtain y from x. It is the **Manhattan** metric between their *representations* p_x and p_x, where $p_z = (p_z(1), \dots, p_z(n))$ and $p_z(i)$ is the number of open parentheses written before the i-th closed parenthese of $z \in P_n$.

  There is a bijection between parentheses strings and binary trees; cf. the **tree rotation distance** in Chap. 15.

  Similarly, *Autord-Dehornoy distance* between shortest expressions x and y of a permutation as a product of transpositions, is the minimal number of *braid relations* needed to get x from y.

- **Schellenkens complexity quasi-metric**
  The **Schellenkens complexity quasi-metric** is a quasi-metric between infinite strings $x = x_0, x_1, \dots, x_m, \dots$ and $y = y_0, y_1, \dots, y_n, \dots$ over $\mathbb{R}_{\geq 0}$ with $\sum_{i=0}^{\infty} 2^{-i} \frac{1}{x_i} < \infty$ (seen as complexity functions), defined (Schellenkens 1995) by

$$\sum_{i=0}^{\infty} 2^{-i} \max\{0, \frac{1}{x_i} - \frac{1}{y_i}\}.$$

- **Graev metrics**
  Let (X, d) be a metric space. Let $\overline{X} = X \cup X' \cup \{e\}$, where $X' = \{x' : x \in X\}$ is a disjoint copy of X, and $e \notin X \cup X'$. We use the notation $(e')' = e$ and $(x')' = x$ for any $x \in X$; also, the letters x, y, x_i, y_i will denote elements of $\overline{X}$. Let $(\overline{X}, D)$ be a metric space such that $D(x, y) = D(x', y') = d(x, y)$, $D(x, e) = D(x', e)$ and $D(x, y') = D(x', y)$ for all $x, y \in X$.

  Denote by $W(X)$ the set of all words over $\overline{X}$ and, for each word $w \in W(X)$, denote by $l(w)$ its length. A word $w \in W(X)$ is called *irreducible* if $w = e$ or $w = x_0 \dots x_n$, where $x_i \neq e$ and $x_{i+1} \neq x'_i$ for $0 \leq i < n$.

  For each word w over $\overline{X}$, denote by $\widehat{w}$ the unique irreducible word obtained from w by successively replacing any occurrence of xx' in w by e and eliminating e from any occurrence of the form w_1ew_2, where at least one of the words w_1 and w_2 is non-empty.

  Denote by $F(X)$ the set of all irreducible words over $\overline{X}$ and, for $u, v \in F(X)$, define $u \cdot v = w'$, where w is the concatenation of words u and v. Then $F(X)$ becomes a group; its identity element is the (non-empty) word e.

  For any two words $v = x_0 \dots x_n$ and $u = y_0 \dots y_n$ over $\overline{X}$ of the same length, let $\rho(v, u) = \sum_{i=0}^{n} D(x_i, y_i)$. The **Graev metric** between two irreducible words $u = u, v \in F(X)$ is defined [DiGa07] by

$$\inf\{\rho(u^*, v^*) : u^*, v^* \in W(X),\ l(u^*) = l(v^*),\ \widehat{u^*} = u, \widehat{v^*} = v\}.$$

  Graev proved that this metric is a **bi-invariant metric** on $F(X)$, extending the metric d on X, and that $F(X)$ is a topological group in the topology induced by it.

11.2 Distances on permutations

A *permutation* (or *ranking*) is any string $x_1 \dots x_n$ with all x_i being different numbers from $\{1, \dots, n\}$; a *signed permutation* is any string $x_1 \dots x_n$ with all $|x_i|$ being different numbers from $\{1, \dots, n\}$. Denote by $(Sym_n, \cdot, id)$ the group of all permutations of the set $\{1, \dots, n\}$, where id is the *identity mapping*.

The restriction, on the set Sym_n of all n-permutation vectors, of any metric on $\mathbb{R}^n$ is a metric on Sym_n; the main example is the l_p**-metric** $(\sum_{i=1}^{n} |x_i - y_i|^p)^{\frac{1}{p}}$, $p \geq 1$.

The main editing operations on permutations are:

- *Block transposition*, i.e., a substring move
- *Character move*, i.e., a transposition of a block consisting of only one character
- *Character swap*, i.e., interchanging of any two adjacent characters