# Алфавітне й рівномірне кодування. Достатні умови однозначності декодування. Властивості роздільних кодів. Оптимальне кодування. Коди, стійкі до перешкод. Коди Хеммінга.

**Алфавітне й рівномірне кодування.** Без утрати загальності можна сформулювати задачу кодування так.

***Означення 1.*** Нехай задано алфавіт $A = \{a_1, \dots, a_n\}$ зі скінченної кількості символів, які називають **буквами**. Скінченну послідовність букв алфавіту *A* називають **словом** у цьому алфавіті. Для позначення слів будемо використовувати малі грецькі літери: $\alpha = a_{i1}a_{i2}\dots a_{il}$. Число $l$ – кількість букв у слові - $\alpha$ називають **довжиною** слова $\alpha$ і позначають $l(\alpha)$ або $|\alpha|$. Множину всіх слів у алфавіті *A* позначають як *A**. Порожнє слово позначають ε; зазначимо, що $\varepsilon \notin A$, $\varepsilon \in A^*$, $|\varepsilon| = 0$.

***Означення 2.*** Якщо слово $\alpha$ має вигляд $\alpha = \alpha_1\alpha_2$, то $\alpha_1$ називають **початком** або **префіксом** слова $\alpha$, а $\alpha_2$ – його **закінченням** або **постфіксом**. Якщо при цьому $\alpha_1 \neq \varepsilon$ (відповідно, $\alpha_2 \neq \varepsilon$), то $\alpha_1$ називають **власним початком** (відповідно власним закінченням) слова $\alpha$.

Для слів визначена операція конкатенації або зчеплення.

***Означення 3.*** **Конкатенація** слів α та β є слово αβ, отримане виписуванням підряд спочатку всіх букв слова α, а потім всіх букв слова β. Зазвичай операція зчеплення ніяк не позначається.

***Означення 4.*** Довільна множина $L$ слів у деякому алфавіті *A* називається **мовою** в цьому алфавіті, $L \subset A^*$.

Нехай задано алфавіти $A = \{a_1, \dots, a_n\}$, $B = \{b_1, \dots, a_m\}$ і функція $F: S \to B^*$, де *S* – деяка множина слів у алфавіті *A*, $S \subset A^*$. Тоді функція *F* називається **кодуванням**, елементи множини *S* – **повідомленнями**, а елементи $\beta = F(\alpha)$, $\alpha \in S$, $\beta \in B^*$ – **кодами** (відповідних повідомлень). Обернена функція $F^{-1}$, якщо вона існує, називається **декодуванням.**

Якщо $|B| = m$, то *F* називають *m*-ковим кодуванням. Найчастіше використовується алфавіт $B = \{0, 1\}$, тобто **двійкове кодування**. Саме його ми й розглядатимемо в наступних підрозділах, випускаючи слово «двійкове».

Відображення *F* задають якимось алгоритмом. Є два способи задати відображення *F*.

1. **Алфавітне кодування** задають схемою (або таблицею кодів) σ:

$$\alpha_1 \to \beta_1,$$
$$\alpha_2 \to \beta_2,$$
$$\dots$$
$$\alpha_n \to \beta_n$$

де $\alpha_i \in A, \beta_i \in B, i = 1, \dots, n$. Схема σ задає відповідність між буквами алфавіту *A* та деякими словами в алфавіті *B*. Вона визначає алфавітне кодування так: кожному слову $\alpha = a_{i1}a_{i2}\dots a_{il}$ із повідомлення *S* поставлено у відповідність слово $\beta = \beta_{i1}\beta_{i2}\dots\beta_{il}$ – його код. Слова $\beta_1, \dots, \beta_n$ називають **елементарними кодами.**

Наприклад, розглянемо алфавіт $A = \{0, 1, 2, 3, 4, 5, 6, 7, 8, 9\}$, $B = \{0, 1\}$ і схему:

$\delta_1 = \langle\ 0 \to 0,\ 1 \to 1,\ 2 \to 10,\ 3 \to 11,\ 4 \to 100,\ 5 \to 101,\ 6 \to 110,\ 7 \to 111,\ 8 \to 1000,\ 9 \to 1001 \rangle$ .

Ця схема однозначна, але кодування не є взаємно однозначним: $F_\delta(333) = 111111 = F_\delta(77)$, а значить, неможливе декодування. З іншого боку, схема $\delta_1 = \langle\ 0 \to 0000,\ 1 \to 0001,\ 2 \to 0010,\ 3 \to 0011,\ 4 \to 0100,\ 5 \to 0101,\ 6 \to 0110,\ 7 \to 0111,\ 8 \to 1000,\ 9 \to 1001 \rangle$ , відома під назвою «двійково-десяткове кодування», допускає однозначне декодування.

2. **Рівномірне кодування з параметрами** *k* й *r* задають так. Повідомлення α розбивають на блоки довжиною *k*:

$$\alpha = (x_1 \dots x_k)(x_{(k+1)} \dots x_{2k}) \dots (x_{(mk+1)} \dots x_{(pk+s)}),$$

де $s \leq k$ (останній блок може бути коротшим, у такому разі спосіб його кодування спеціально обумовлюють), $x_i \in A, i = 1, \dots, pk + s$. Блоки довжиною *k* розглядають як «букви» якогось алфавіту (таких блоків, очевидно, $n^k$, бо алфавіт *A* складається з *n* букв) і кодують словами в алфавіті *B* довжиною *r* за **схемою рівномірного кодування** $\delta_{(k,r)}$:

$$\alpha_1 \to \beta_1,$$
$$\alpha_2 \to \beta_2,$$
$$\dots$$
$$\alpha_{(n^k)} \to \beta_{(n^k)}$$

**Надлишковістю схеми $\delta_{(k,r)}$ на символ повідомлення** називають величину $R = (n-k)/k = (n/k) - 1$.

**Достатні умови однозначності декодування. Властивості роздільних кодів.**

Розглянемо схему алфавітного кодування $\delta$ та різні слова, складені з елементарних кодів. Схему $\delta$ називають **роздільною**, якщо з рівності $\beta_{i1}\dots\beta_{ik} = \beta_{j1}\dots\beta_{jl}$ випливає, що $k=l$ та $i_t = j_t$ для кожного $t$=1, …, $k$, тобто будь-яке слово, складене з елементарних кодів, можна єдиним способом розкласти на елементарні коди. Очевидно, що алфавітне кодування з роздільною схемою вможливлює однозначне декодування.

Схему $\delta$ називають **префіксною**, якщо для будь-яких $i$ та $j$ ($i,j$ = 1, …,$r$; $i \neq j$) елементарний код $\beta_i$ не є префіксом елементарного коду $\beta_j$ .

***Теорема 1.*** Префіксна схема є роздільною.

***Доведення***. Доведемо цю теорему від протилежного. Оскільки схема $\delta$ префіксна, то всі елементарні коди в ній попарно різні, тобто $\beta_i \neq \beta_j$, якщо $i \neq j$. Нехай кодування зі схемою $\delta$ не роздільне. Тоді існує таке слово $\beta \in B$, що можуть бути два розклади на елементарні коди: $\beta = \beta_{i1}\dots\beta_{ik} = \beta_{j1}\dots\beta_{jl}$.

Нехай $\beta_{i1} = \beta_{j1}, \dots, \beta_i(p-1) = \beta_j(p-1)$ але $\beta_{ip} \neq \beta_{jp}$ . У такому разі одне зі слів $\beta_{ip}, \beta_{jp}$ являє собою префікс іншого, а це суперечить тому, що схема $\delta$ префіксна.

Властивість префіксності достатня але не необхідна умова роздільності схеми.

Приклад 1. Нехай $A = \{a, b\}, B = \{0,1\}, \delta: a \to 0, b \to 01$. Схема $\delta$ не префіксна, але роздільна. Справді, перед кожним входженням 1 в слові $\beta_2$ безпосередньо є 0. Це дає змогу виділити всі пари (01). Частина слова, що залишилась, складається із символів 0.

Нехай $\beta = \beta_{i1}\beta_{i2}\dots\beta_{ik}$ - слово з $B$. Позначимо як $\tilde{\beta}$ слово, одержане оберненням слова $\beta$ тобто $\tilde{\beta} = b_{(ik+1)}b_{ik}\dots b_{i1}$. Позначимо як $\tilde{\sigma}$ схему:

$$\alpha_1 \to \widetilde{\beta_1},$$
$$\alpha_2 \to \widetilde{\beta_2},$$
$$\dots$$
$$\alpha_n \to \widetilde{\beta_n}$$

Приклад 2. Візьмемо схему $\sigma$ з прикладу 1. Тоді $\tilde{\sigma}$ має такий вигляд: $a \to 0, b \to 10$. Схема $\tilde{\sigma}$ префіксна, тому за теоремою 1 вона роздільна. Зауваження. Схеми $\sigma$ та $\tilde{\sigma}$ водночас або роздільні, або ні. Це зауваження дає змогу посилити теорему 1.

***Теорема 2.*** Якщо або схема $\sigma$, або схема $\tilde{\sigma}$ префіксна, то обидві схеми $\sigma$ та $\tilde{\sigma}$ роздільні.

Можна навести приклад такої роздільної схеми $\sigma$, що ні $\sigma$, ні $\tilde{\sigma}$ не префіксні. Отже, теорема 2 також дає достатню, але не необхідну умову роздільності схеми. Нехай задано схему алфавітного кодування $\sigma$: $a_i \to \beta_i, a_i \in A, \beta_i \in B, i = 1, \dots, r$.

Для того, щоб схема алфавітного кодування була роздільною, необхідно, щоб довжини елементарних кодів задовільняли **нерівність Мак-Міллана** (B. McMillan).

У **теорії кодування** нерівність Крафта - Макміллана дає **необхідну і достатню умову** існування **роздільних** і **префіксних** кодів з заданим набором довжин кодових слів.

Нехай задано дві довільно кінцеві множини, які називаються, відповідно, кодованим алфавітом і кодуючим алфавітом. Їх елементи називаються символами, а рядки (послідовності кінцевої довжини) символів — словами. Довжина слова — це число символів, з якого воно складається.

У якості кодуючого алфавіту часто розглядається множина *{0;1}* — так званий двійковий або бінарний алфавіт.

Схемою алфавітного кодування (або просто (алфавітним) кодом) називається будь-яке відображення символів кодованого алфавіту в слова кодуючого алфавіту, які називають кодовими словами. Користуючись схемою кодування, кожне слово кодованого алфавіту можна зіставити з його

кодом — конкатенацією кодових слів, відповідних кожному символу цього слова.

Код називається роздільним (або однозначно декодованим), якщо жодним двом словам кодованого алфавіту не може бути зіставлений один і той же код. Префіксним кодом називається алфавітний код, в якому жодне з кодових слів не є префіксом жодного іншого кодового слова. Будь-який префіксний код є роздільним.

***Теорема 3.(нерівність Мак-Міллана).*** Якщо схема алфавітного кодування роздільна, то $\sum_{i=1}^{r} 1/2^{li}$,де $l_i = l(\beta_i)$.

***Теорема 4.*** Якщо числа $l_1, \dots, l_n$задовольняють нерівність

$$\frac{\sum_{i=1}^{r} 1}{2^{li}} \leqslant 1$$(нерівність Мак-Міллана),

то існує префіксна схема алфавітного кодування $\sigma$:

$$\begin{matrix} a_1 \to \beta_1 \\ \dots\dots\dots \\ a_r \to \beta_r \end{matrix},$$

де $l_1 = l(\beta_1)$,..., $l_r = l(\beta_r)$.

Розглянемо слова довжиною $\lambda_1$ у двійковому алфавіті *B*. Оскільки $v_1 \leqslant 2^{(\lambda_1)}$, то мижна вибрати з них $v_1$різних слів $\beta_1$,..., $\beta_{v1}$довжиною $\lambda_1$. Вилучимо з подальшого розгляду всі слова з множини *B*, які починаються зі слів $\beta_1$,..., $\beta_{v1}$. Потім розглянемо множину слів у алфавіті *B*, що мають довжину$\lambda_2$та не починаються зі слів $\beta_1$,..., $\beta_{v1}$. Таких слів, очевидно, $2^{(\lambda_2)} - v_1 2^{(\lambda_2 - \lambda_1)}$. Позаяк $v_2 \leqslant 2^{(\lambda_2)} - v_1 2^{(\lambda_2 - \lambda_1)}$, то з цієї множини можна вибрати $v_2$ різних слів; позначимо їх як $\beta_{(v_1+1)}, \dots, \beta_{(v_1+v_2)}$. Вилучимо з множини $B$ слова, які починаються з $\beta_{(v_1+1)}, \dots, \beta_{(v_1+v_2)}$, з подальшого розгляду. Продовжимо цей процес, поступово використовуючи допоміжні нерівності. Одержимо слова $\beta_1$,..., $\beta_r$, де $r = \sum_{i=1}^{\mu} v_i$. Узявши їх як елементарні коди , отримаємо якусь схему алфавітного кодування $\sigma$. Вона, очевидно, префіксна. Крім того, $l(\beta_1) = l_1$,..., $l(\beta_r) = l_r$.

***Наслідок 1.*** Нерівність Мак-Міллана — необхідна й достатня умова імнування алфавітного кодування з префіксною схемою та довжинами елементарних кодів, що дорівнюють $l_1$,..., $l_r$.

***Наслідок 2.*** Якщо існує роздільна схема алфавітного кодування із заданими довжинами елементарних кодів, то існує й префіксна схема з тими самими довжинами елементарних кодів.

Для доведення потрібно спочатку застосувати теорему 3, а потім — теорему 4.

Досі термін "код" було використано в загальноприйнятому розумінні. Однак у теорії кодування,а ще раніше в техніці слово "код" трактують також і як множину елементарних кодів. Починаючи з цього місця, будемо використовувати слово "код" у двох значеннях. З контексту буде зрозуміло,яке саме з них ми маємо на увазі.

Нехай задано алфавіт $A = \{a_1, \dots, a_r\}$і ймовірності $P = (p_1, \dots, p_r)$появи букв у повідомленні; тут$p_1$- ймовірність появи букви$a_1$.Не зупиняючись на означенні ймовірності, яких вона з'являється, від загальної кількості букв у повідомленні. Так, якщо$p_1 = 0.25$, то це означає , що з, наприклад, 1000 переданих букв близько 250 разів з'явиться буква$a_1$.Не втрачаючи загальності, можна вважати, що

$$p_1 \geq p_2 \geq \dots \geq p_r > 0$$

тобто можна одразу вилучити букви, які не можуть з`явитися у повідомленні, і впорядкувати букви за спаданням ймовірностей їх появи. Крім того, $p_1 + p_2 + \dots + p_r = 1$. Кортеж $P = (p_1, \dots, p_r)$ називають *розподілом ймовірностей* (появи букв алфавіту А).

Для кожної роздільної схеми алфавітного кодування σ математичне сподівання коефіцієнта збільшення довжини повідомлення в разі кодування(позначають$l_{\text{сер}}^{\sigma}$) визначають так:

$$l_{\text{сер}}^{\sigma}(P) = \sum_{i=1}^{r} p_i l_i$$

де $l_i = l(B_i), i = 1, \ldots, r$,і називають середньою довжиною кодування за схемою σ для розподілу ймовірностей Р.

Очевидно, що величина$l^{\sigma}_{\text{сер}}(l^{\sigma}_{\text{сер}} > 1)$показує, у скільки разів зросте середня довжина слова у разі кодування зі схемою σ.

Уведемо величину $l_a = inf\, l^{\sigma}_{\text{сер}}$, де нижню грань беруть за всіма роздільними схемами σ. Легко побачити, що

$$1 \le l_a \le \lceil \log r \rceil$$

де верхню оцінку дає схема з елементарними кодами з однаковою довжиною $\lceil \log r \rceil$ (логарифм береться з основою 2).Звідси випливає, що для побудови кодів, у яких величина $l_{\text{сер}}$ близька до$l_a$, можна не враховувати коди з більшим $l_{\text{сер}}$,ніж$\lceil \log r \rceil$. Отже, будемо розглядати лише схеми з $p_i l_i \le \lceil \log r \rceil$.Позначимо $p_a = min(p_1, \ldots, p_r)$,тоді

$$l_i \le \lceil \log r \rceil / p$$

для всіх $i = 1,,2, \ldots, r$.

Звідси випливає, що є лише скінченна кількість варіантів значень$l_{\text{сер}}$, для яких$l_a \le l_{\text{сер}} \le \lceil log r \rceil$.Отже,значення$l_a$досягається на якійсь схемі σ; його можна визначити як $l_a = min l^{\sigma}_{\text{сер}}$.

Коди, визначені схемою σ з $l_{\text{сер}} = l_a$,називають кодами з мінімальною надлишковістю(або оптимальними кодами) для розподілу ймовірностей Р. Існує алфавітне кодування з префіксною схемою, яке дає оптимальні коди. У зв`язку з цим, будуючи оптимальні коди, можна обмежитися префіксними схемами.

Проста ідея побудови коду, близького до оптимального, належить американському математикові Р. Фано (R.Fano). Сформулюємо алгоритм кодування за методом Фано.

1. Упорядковуємо букви алфавіту А за спаданням імовірностей їх появи в повідомленні.

2. Розбиваємо множину букв, записаних у зазначеному порядку, на дві послідовні (тобто без перестановок букв) частини так, щоб сумарні ймовірності кожної із них були якомога близькими одна до одної. Кожній букві з першої частини приписуємо символ 0, другої - символ 1. Далі те саме робимо із кожною частиною, якщо вона містить принаймні дві букви. Процедуру продовжуємо доти, доки всю множину не буде розбито на окремі букви.

**Приклад 1.** Нехай задано розподіл імовірностей P= (0.4, 0.15, 0.15, 0.15, 0.15). Побудуємо код за методом Фано. Розв'язок подано в таблиці 1 .

Середня довжина коду дорівнює

$$l^{F}_{\text{сер}} = 0.4\text{x}2 + 0.15\text{x}2 + 0.15\text{x}3 + 0.15\text{x}3 = 2.3$$

тут верхній індекс F означає, що код отримано методом Фано.

| Букви алфавіту А | Імовірності появи букв | Розбиття множини букв | | | Елементарні коди |
|---|---|---|---|---|---|
| $a_1$ | 0.4 | 0 | 0 | | 00 |
| $a_2$ | 0.15 | | 1 | | 01 |
| $a_3$ | 0.15 | 1 | 0 | | 10 |
| $a_4$ | 0.15 | | 1 | 0 | 110 |
| $a_5$ | 0.15 | | | 1 | 111 |

Алгоритм Фано має просту інтерпретацію за допомогою бінарного дерева. Від кореня відходять два ребра, ліве позначено символом 0, а праве — символом 1. Ці два ребра відповідають розбиттю множини всіх букв на дві майже рівномірні частини, одній з яких поставлено у відповідність символ 0, а другій — символ 1.Ребра, що виходять із вершин наступного рівня, відповідають розбиттю одержаних частин знову на дві майже рівноймовірні послідовні частини. Цей процес продовжують доти, доки множину букв не буде розбито на окремі букви. Кожний листок

дерева відповідає певному елементарному коду. Щоб виписати цей код, потрібно пройти шлях від кореня до відповідного листа.

Зазначимо, що незалежно від способу кодування кожному бінарному дереву відповідає набір двійкових елементарних кодів. У такому разі дерево називають кодовим. Якщо елементарні коди відповідають листкам кодового дерева, то відповідна схема алфавітного кодування є префіксною(отже, забезпечується однозначністю декодування).

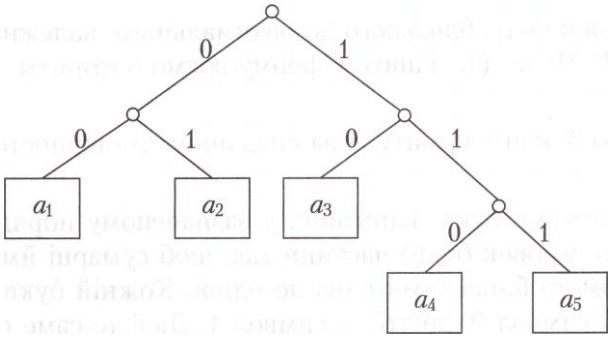


Рис. 1.

**Приклад 2.** На рис. 1. зображено кодове дерево, отримане методом Фано для розподілу ймовірностей із прикладу 1.

Що імовірніша поява букви, то швидше вона утворить “самостійну” підмножину, і тому її буде закодовано коротшим елементарним кодом. Тому метод Фано доволі ефективний. Але чи завжди він дає змогу отримати оптимальний код? Виявляється, що ні.

Один з перших алгоритмів ефективного кодування інформації був запропонований Д. А. Хаффманом в 1952 році. Ідея алгоритму полягає в наступному: знаючи ймовірності символів у повідомленні, можна описати процедуру побудови кодів змінної довжини, що складаються з цілої кількості бітів. Символам з більшою ймовірністю ставляться у відповідність більш короткі коди. Коди Хаффмана володіють властивістю префіксності (тобто жодне кодове слово не є префіксом іншого), що дозволяє однозначно їх декодувати.

Класичний алгоритм Хаффмана на вході отримує таблицю частот з якими зустрічаються символи у повідомленні. Далі на підставі цієї таблиці будується дерево кодування Хаффмана (Н-дерево).

1. Символи вхідного алфавіту утворюють список вільних вузлів. Кожен лист має вагу, яка може бути рівною або ймовірності, або кількості входжень символу у стиснене повідомлення.
2. Вибираються два вільних вузла дерева з найменшими вагами.
3. Створюється їх батьковий вузол з вагою, рівною їх сумарній вазі.
4. Вузол-батько додається в список вільних вузлів, а два його нащадка видаляються з цього списку.
5. Одній дузі, котра виходить з вузла батька, ставиться у відповідність біт 1, інший - біт 0.
6. Кроки, починаючи з другого, повторюються до тих пір, поки в списку вільних вузлів не залишиться тільки один вільний вузол. Він і буде вважатися коренем дерева.

Припустимо, у нас є наступна таблиця частот:

| | | | | |
|---|---|---|---|---|
| 15 | 7 | 6 | 6 | 5 |
| A | B | C | D | E |

Цей процес можна представити як побудова дерева, корінь якого - символ з сумою ймовірностей об'єднаних символів, що вийшов при об'єднанні символів з останнього кроку, його n0нащадків - символи з попереднього кроку і т. д.

Щоб визначити код для кожного із символів, що входять в повідомлення, ми повинні пройти

шлях від листа дерева, який відповідає поточному символу, до його кореня, накопичуючи біти при переміщенні по гілках дерева (перша гілка в дорозі відповідає молодшому біту). Отримана таким чином послідовність бітів є кодом даного символу, записаним у зворотному порядку.

Для даної таблиці символів коди Хаффмана будуть виглядати наступним чином.

| A | B | C | D | E |
|---|---|---|---|---|
| 0 | 100 | 101 | 110 | 111 |

Оскільки жоден з отриманих кодів не є префіксом іншого, вони можуть бути однозначно декодовані при читанні їх з потоку. Крім того, найбільш частий символ повідомлення А закодований найменшою кількістю біт, а найбільш рідкісний символ Д - найбільшим.

При цьому загальна довжина повідомлення, що складається з наведених у таблиці символів, складе 87 біт (в середньому 2,2308 біта на символ). При використанні рівномірного кодування загальна довжина повідомлення склала б 117 біт (рівно 3 біта на символ). Зауважимо, щоентропія джерел, незалежним чином породжує символи із зазначеними частотами, складає ~ 2,1858 біта на символ, тобто надмірність побудованого для такого джерела коду Хаффмана, що розуміється, як відмінність середнього числа біт на символ від ентропії, становить менше 0,05 біт на символ.

**Коди, стійкі до перешкод. Коди Хеммінга.** Коди Хеммінга – лінійні коди, які забезпечують виявлення та корекцію помилок. Використовуються при передачі та зберіганні даних. Особливістю даного коду є використання кількох бітів контролю парності. Коди Хеммінга забезпечують виявлення двох помилок і виправлення однієї помилки. Розглянемо один частковий випадок рівномірного двійкового кодування, коли A=B={0,1}. Розглянемо схему рівномірного кодування $\sigma_{k.n}$із параметрами k,n:

$$\alpha_1 \to \beta_1,$$
$$\alpha_2 \to \beta_2,$$
$$\alpha_3 \to \beta_3,$$
$$\dots\ \dots\ \dots$$
$$\alpha_{(2^k)} \to \beta_{(2^k)},$$

де $\alpha_l, \beta_i$ - відповідно слова довжиною k та n, n>k. Говорять, що схему $\sigma_{k.n}$ задає код $V = \left(\beta_1, \beta_2, \beta_3, \dots, \beta_{(2^k)}\right) \subset E_2^n$. Слова в алфавіті {0,1} – це впорядковані набори з нулів і одиниць (їх називають також двійковими векторами). Зазначимо, що кількість усіх слів$\alpha_l$ довжиною k в алфавіті {0,1} дорівнює $2^k$.

Із методичних міркувань у цьому підрозділі нам зручно використовувати такі позначення. Елементи множини $E_2^n$ (двійкові вектори довжиною n) позначатимемо великими латинськими буквами X, Y, Z, …, залежно від контексту.

Нормою ||X|| двійкового вектора $X = x_1x_2...x_n$називають число, яке дорівнює кількості його одиничних компонентів. Отже, $(X) = \sum_{i=1}^{n} \quad x_i$.

Коди, що самоконтролюються, дозволяють автоматично виявляти найбільш ймовірні помилки при передачі даних. Для їх побудови досить приписати до кожного слова один додатковий (контрольний) двійковий розряд і вибрати цифру цього розряду так, щоб загальна кількість одиниць в зображенні будь-якого числа була, наприклад, парною. Одиночна помилка в довільному розряді переданого слова (зокрема, можливо, і в контрольному розряді) змінить парність загальної кількості одиниць. Лічильники по модулю 2, що підраховують кількість одиниць, які містяться серед двійкових цифр числа, можуть давати сигнал про наявність помилок.

При цьому, зрозуміло, ми не отримуємо ніяких вказівок про те, в якому саме розряді відбулася помилка, і, отже, не маємо можливості виправити її. Залишаються непоміченими також помилки, які виникають одночасно в двох, в чотирьох або взагалі в парній кількості розрядів. Втім, подвійні, а тим більше чотирикратні помилки вважаються малоймовірними.

Припустимо, що в каналі зв'язку діє джерело адитивних перешкод, яке описують множиною P(n,t). Її елементи — двійкові вектори-помилки $x_1x_2...x_s$, у яких норма будь-якого фрагмента $x_lx_{(i+1)} \dots x_{(l+L-1)}$не більша ніж t, якщо довжина фрагмента $L \leqslant n$, (тобто на n переданих поспіль двійкових символів припадає не більше ніж t помилок). Це означає, що коли на вході каналу зв'язку

передано повідомлення a, то на виході можна отримати будь-яке слово з множини $\{a \oplus \gamma \mid \gamma \in P(n,t), l(a) = l(\gamma) \mid\}$ , де $a \oplus \gamma$ - порозрядне додавання за mod 2. Позаяк проблема локалізації інформації (тобто розділення закодованого повідомлення на елементарні коди) у моделі рівномірного кодування тривіальна, то виявлення помилок полягає у відшуканні незбігу локалізованої групи n символів ні з яким елементарним кодом. Якщо через помилку елементарний код перейде в інший елементарний код, то помилку не буде виявлено. Іноді можна виправити помилку. Якщо групу бітів локалізовано правильно, то для нього необхідно й достатньо, щоб помилкова група була "синонімом" єдиного елементарного коду.

Канал зв'язку називають надійним, якщо будь-які помилки можна виявити чи виправити відповідно до заданої мети декодування. Далі наведено головні положення побудови кодів, які забезпечують надійність найпростіших каналів зв'язків.

Віддаллю Хеммінга називають функцію p(X,Y) двох змінних, означену на множині $E_2^n$: $\rho(X,Y) = \sum_{i=1}^{n}(x_i \oplus y_i)$(вона дорівнює кількості розрядів, у яких вектори X та Y не збігаються).

$$\rho(X \oplus Z, Y \oplus Z) = \rho(X,Y),$$
$$\rho(X,Y) = (X) + (Y) - 2(X,Y)$$

**Скалярний добуток** векторів $X, Y \in E_2^n$означають так:

$$\langle X,Y\rangle = \sum_{i=1}^{n} x_i y_i$$

він дорівнює кількості розрядів, у яких X та Y збігаються й дорівнюють 1. Легко перевірити такі співвідношення:

$$\rho(X,0) = (X) = \sum_{i=1}^{n} \quad x_i,$$

де 0 — n-вимірний вектор із нульовими компонентами;

$$\rho(X,Y) = (X \oplus Y),$$

де $X \oplus Y$- порозрядне додавання за mod 2;

Для віддалі Хеммінга виконуються аксіоми метрики:

- $\rho(X,Y) \geqslant 0$, причому $\rho(X,Y) = 0$в тому й лише в тому разі, якщо $X=Y$;
- $\rho(X,Y) = \rho(Y,X)$;
- $\rho(X,Y) + \rho(Y,Z) \geq \rho(X,Z)$(нерівність трикутника).

Метрика Хеммінга — зручне математичне поняття для формулювання умов дійсності кодування в разі адитивних помилок. Нехай схему $\sigma_{(k,n)}$ задано кодом $V = \{\beta_1, \beta_2\beta_3,..., \beta_{(2^k)}\}$. Кодовою віддалю для коду V називають величину

$\rho(V) = min\{\rho(X,Y) \mid X,Y \in V, X \neq Y\}$.

**Теорема 4.** Якщо в каналі зв'язку діє джерело адитивних перешкод P(n,t), то правдиві такі твердження:

1. Для виявлення будь-яких помилок необхідно й достатньо, щоз p(V)>t.
2. Для виправлення будь-яких помилок необхідно й достатньо, щоб p(V)>2t.

**Зауваження.** Інакше кажучи, код може виявляти будь-які комбінації з t й меншої кількості помилок тоді й лише тоді, коли його кодова віддаль більша ніж t;

Код може виправляти будь-які комбінації з t й меншої кількості помилок тоді й лише тоді, коли його кодава віддаль більша ніж 2t.

**Теорема 5.** Якщо в каналі зв'язку діє джерело адитивних перешкод P(n,t), то правдиві такі твердження:

1. Для виявлення будь-яких помилок необхідно й достатньо, щоб для будь-якого $X \in V$куля $S_t(X)$ не містила інших елементарних кодів, окрім X.

2. Для виправлення будь-яких помилок необхідно й достатньо, щоб для будь-яких $X, Y \subset V$ було виконано умову $S_t(X) \cap S_t(Y) = \emptyset$ .

Рівномірне кодування $\sigma_{(k,n)}: a_i \to \beta_i (i = 1,2,3,..., 2^k)$ називають систематичним, якщо можна виділити множину k розрядів $I = \{i_1,..., i_k\} \subset \{1,2,..., n\}$, які називають інформаційними, так, що коли $\beta_i = x_1...(i = 1,2,3,..., 2^k)$, то $a_i = x_{(i_i)}...x_{(i_k)}$. Решта розрядів у такому разі називають контрольними.