Лекція 4. Теоретичні основи кодування.

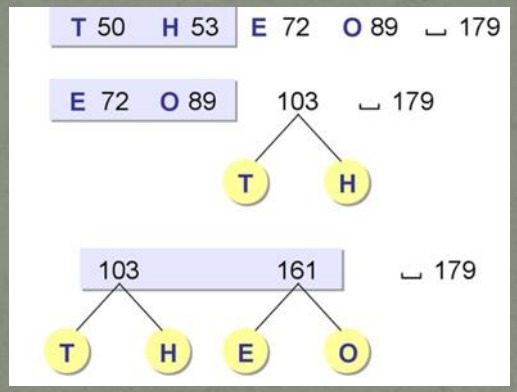
Вступ

1. Загальні положення. Класифікація кодів.
2. Ефективне кодування.



	%		%		%
A	8.1	K	0.4	U	2.4
B	1.4	L	3.4	V	0.9
C	2.7	M	2.5	W	1.5
D	3.9	N	7.2	X	0.2
E	13.0	O	7.9	Y	1.9
F	2.9	P	2.0	Z	0.1
G	2.0	Q	0.2		

алгоритм стиснення даних Шеннона - Фано



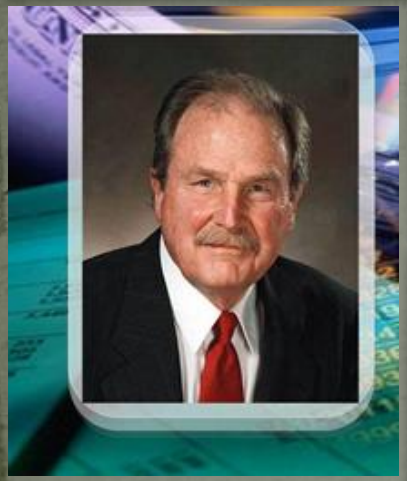


Девід Хаффман

Ю. Бродський (кафедра КБ та ЗІ)

Роберт Маріо Фано (1917 Турин, Італія — 2016, Нейплз, США) Італійсько - американський вчений, почесний професор з електроніки і комп'ютерних наук в Массачусетському технологічному інституті. Відомий своїми роботами з теорії інформації, разом з Клодом Шеннономом розробив алгоритм Шеннона-Фано. Працював директором Массачусетського технологічного інституту з моменту його заснування в 1963 році до 1968 року.

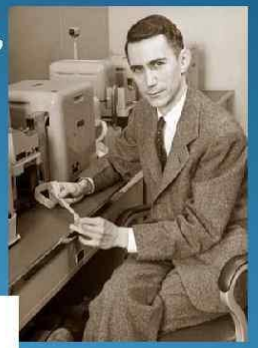


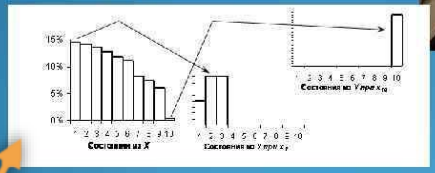


Девід Гаффман (англ. David Albert Huffman, 9 серпня 1925, Альянс, США — †7 жовтня 1999, Санта-Круз, США)

У 1952 створив алгоритм префіксного кодування з мінімальною надмірністю (відомий як алгоритм або код Гаффмана).

В 1937 Клод Шеннон предствить до захисту кандидатську дисертацію Символічний аналіз релейних і перемикальних схем в MIT, в якій булева алгебра і двійкова арифметика були використані стосовно до електронних реле і перемикачів. На дисертації Шеннона по суті заснована вся сучасна цифрова техніка.

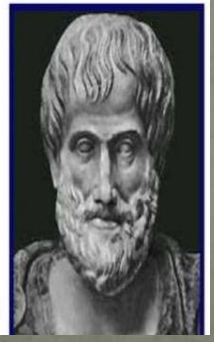




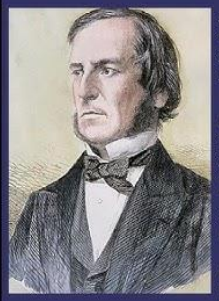
АРІСТОТЕЛЬ

(384 р.-322 р. до н.е.)

- Заклав основи формальної логіки
- Виклав закони логічного виведення
- Розробив аксіоматичний метод
- Запропонував першу формально-аксіоматичну систему логіки - **силогістику**
- Заклав основи модальної логіки



Алгебра логіки



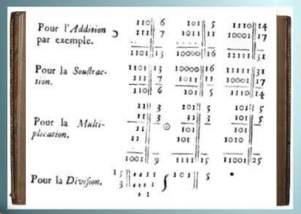
Алгебра логіки (*Булева алгебра, Булева логіка, двійкова логіка, двійкова алгебра*, англ. *Boolean algebra*) — розділ математичної логіки, що вивчає систему логічних операцій над висловлюваннями.

Булеву алгебру запропонував **Джордж Буль** у своїй книзі *Математичний аналіз логіки* **(1847)**, і більш детально у наступній книзі *Дослідження законів думки* (1854).Відповідно до Гантінгтона, термін «Булева алгебра» вперше запропонував Шеффер в 1913, хоча Чарлз Сандерс Пірс в 1880 дав назву «Булева алгебра із однією сталою» першій главі своєї книги «Найпростіша математика».

Булева алгебра є фундаментальною основою для розвитку цифрової електроніки, і втілена в усіх мовах програмування. Вона також використовується у теорії множин і статистиці.



Сучасна двійкова система була повністю описана Лейбніцем в XVII столітті в роботі Explication de l'Arithmtique Binaire. В системі числення Лейбніца були використані цифри 0 і 1, як і в сучасній двійковій системі.





Електронно-обчислювальні машини

Клод Шеннон

Принципи двійкової алгебри були вперше сформульовані математиком ***Джорджем Булєм*** в середині XIX століття, але тільки ***Клод Шеннон*** визначив, як застосувати цей вид логіки до електронних комунікацій.

У 30-х роках XX ст., коли він ще був студентом ***Ванневара Буша***, Клод Шеннон розробив цифровий аналізатор, мабуть, найвидатніший в історії механічний (аналоговий) калькулятор. У статті "Символьний аналіз релейних та комутаційних ланцюгів", яка поклала початок довгому співробітництву з Bell Laboratories, Шеннон сформулював основи теорії взаємодії між символьною логікою та релейними ланцюгами.

Працюючи в Bell Labs, Шеннон написав фундаментальну роботу "Математична теорія комунікацій". *Інформацію, що міститься в повідомленні, завжди можна передати у вигляді послідовності нулів та одиниць, питання тільки в тому, скільки таких елементів інформації потрібно.*

Клод Шеннон (сумісно з Н.Вінером) визначив поняття *біта*.



Клод Шеннон (1916-2001)

1. Загальні положення. Класифікація кодів.

Мета кодування – пристосувати форму повідомлення до характеристик даного каналу зв'язку чи пристрою, призначеному для перетворення або збереження інформації.

Кодування в системах передачі інформації (СПІ) – це відображення множини дискретних повідомлень на множину елементів (літер), які у свою чергу відображаються елементарними сигналами переносниками інформації.

Множини елементів називаються ***кодами*** або ***кодовими комбінаціями*** повідомлень, при цьому кожному повідомленню відповідає свій код.

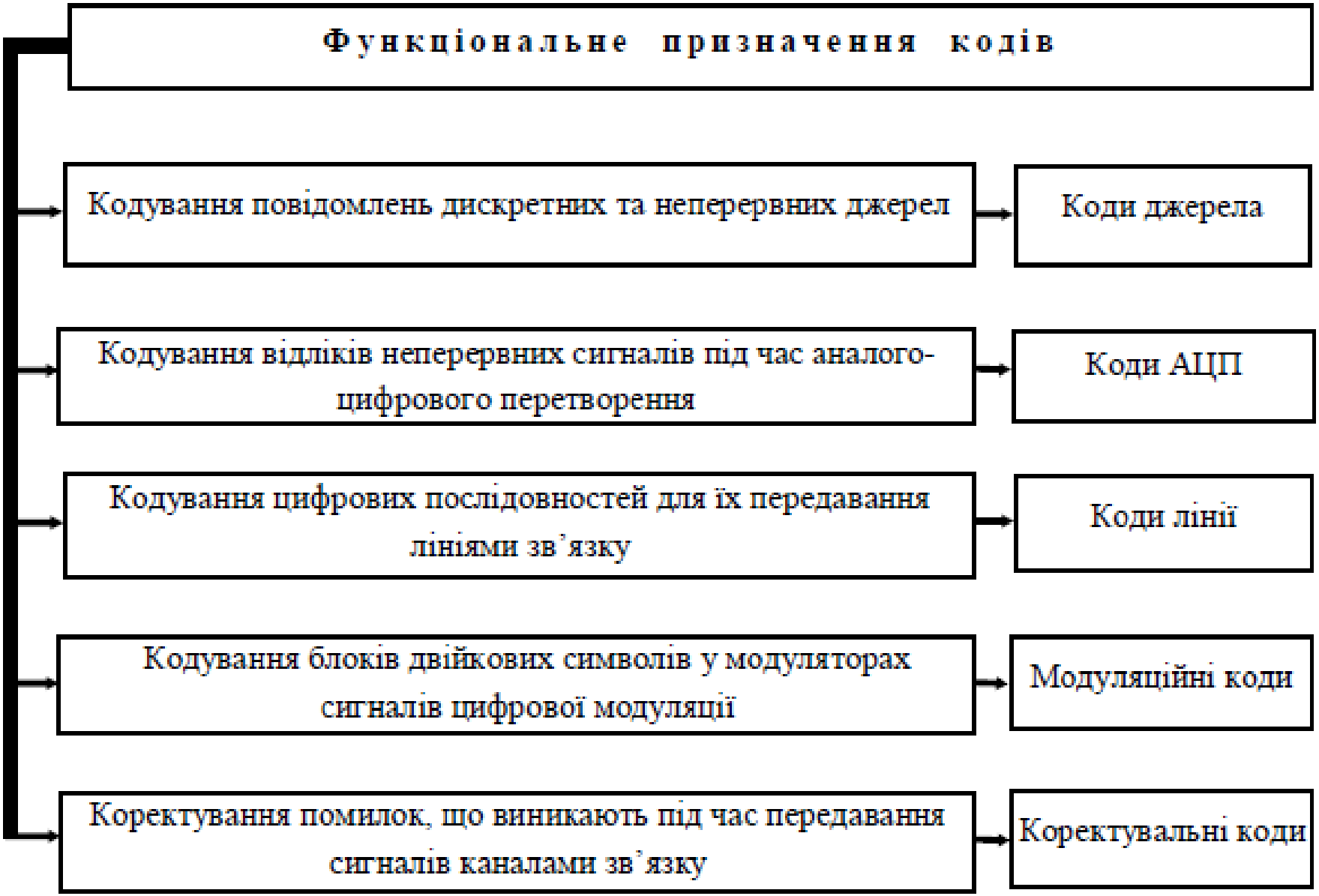
Кодом також називають правила, за якими утворюються кодові комбінації. Кодування може виражатися також у переході від одного коду до іншого, побудованому за своїми власними правилами (наприклад, від нормального двійкового до двійково-десяткового).

Кодування повідомлень може переслідувати цілі:

а) одержання необхідної множини повідомлень при обмеженій кількості використовуваних елементів (об'ємі алфавіту, мінімум $m = 2$);

б) досягнення максимальної ефективності СПІ шляхом зниження надлишковості переданих повідомлень;

в) забезпечення необхідної вірогідності передачі інформації в каналах з перешкодами.


Функціональне призначення кодів
Кодування повідомлень дискретних та неперервних джерел
Коди джерела
Кодування відліків неперервних сигналів під час аналого-цифрового перетворення
Коди АЦП
Кодування цифрових послідовностей для їх передавання лініями зв'язку
Коди лінії
Кодування блоків двійкових символів у модуляторах сигналів цифрової модуляції
Модуляційні коди
Коректування помилок, що виникають під час передавання сигналів каналами зв'язку
Коректувальні коди

Примітивні коди

Натуральні коди – це подання числового номера знака (перед кодуванням знаки джерела необхідно певним чином перенумерувати) у будь-якій позиційній системі числення з основою m.

Знаки повідомлень	Код з $m = 10$	Код з $m = 8$	Код з $m = 4$	Код з $m = 2$
a_0	0	00	00	0000
a_1	1	01	01	0001
a_2	2	02	02	0010
a_3	3	03	03	0011
a_4	4	04	10	0100
a_5	5	05	11	0101
a_6	6	06	12	0110
a_7	7	07	13	0111
a_8	8	10	20	1000

Стандартні коди – це подання знаків джерела стандартизованими кодовими комбінаціями.

Стандартизація провадиться національними або міжнародними організаціями зі стандартизації.

Перелік стандартних кодів за більш як 170-річну історію існування електричного зв'язку досить великий.

До них відносяться коди:

- Морзе;
- Бодо (перший міжнародний п'ятирозрядний телеграфний код);
- МТА № 2 (Міжнародний телеграфний алфавіт № 2);
- МТК № 2 (Міжнародний телеграфний код № 2, який відрізняється від МТА № 2 наявністю кирилиці);
- МТА № 3; МТК № 5;
- російські – КОИ-7 (код відображення інформації); ДКОИ-8 (двійковий код відображення інформації);
- американські – ASCII (American Standard Code for Information Interchange – американський стандартний код для обміну інформацією) та EBCDIC (Extended Binary Coded Decimal Interchange) – розширений двійково - десятковий код обміну інформацією тощо.

2. Ефективне кодування

Ефективними (економними) називають коди джерел повідомлень, які забезпечують зменшення надмірності повідомлень під час кодування. Завдяки зменшенню надмірності повідомлень ефективні коди надають можливість більш ефективно використовувати канали зв'язку чи пристрої пам'яті.

Ефективне кодування повідомлень в обчислювальній техніці називають також ***стисненням інформації***.

Оскільки для побудови ефективного коду використовують імовірності знаків, то ефективні коди називають також *статистичними.*

Для того, щоб сформулювати принципи побудови ефективних кодів, необхідно згадати причини надмірності повідомлень:

- ✓ статистична залежність між знаками в повідомленнях джерела;
- ✓ нерівноймовірність знаків у повідомленнях джерела.

Принципи побудови ефективних кодів

Ю. Бродський (кафедра КБ та ЗІ)

Принципи побудови ефективних кодів

І етап: усувається статистична залежність між знаками, що підлягають кодуванню, шляхом укрупнення алфавіту. Укрупнення алфавіту полягає в тому, що деякі знаки джерела *A* об'єднуються в знакосполучення з кількох знаків. Об'єднання виконується так, щоб знакосполучення в повідомленнях були незалежними. Якщо знаки джерела *A* незалежні, то перший етап не потрібний.

ІІ етап - необхідно врахувати наступні особливості:

- зменшення надмірності можливе лише при використанні нерівномірних кодів;
- чим більша ймовірність знака, тим менше двійкових символів (меншу довжину) повинна мати кодова комбінація,
- щоб ентропія повідомлення на виході кодера була близька до максимального значення (1 біт), ймовірності символів „1“ і „0“ на виході кодера повинні бути близькими до 0,5;
- код повинен бути ***префіксним*** – *ніяка з коротких кодових комбінацій не повинна бути початком більш довгої кодової комбінації.* Тоді для правильного декодування послідовності не потрібно передавати розділові знаки між кодовими комбінаціями.

Ю. Бродський (кафедра КБ та ЗІ)

Числові характеристики ефективного коду

Найменування параметра	Розрахункова формула
Кількість кодових комбінацій рівномірного коду	$M = m^n$
Довжина (розрядність) рівномірного двійкового коду	$n = \log_2 M$
Співвідношення між обсягом алфавіту (кількістю знаків) джерела та кількістю кодових комбінацій коду	$M_A \le M$
Тривалість двійкового символу (біта)	$T_б = T_{зн}/n$
Середня довжина кодових комбінацій нерівномірного коду	$\overline{n} = \sum_k P(a_k) n_k$
Коефіцієнт стиснення нерівномірного коду η	$\eta = n/\overline{n}$
Коефіцієнт ефективності нерівномірного коду μ	$\mu = H(A)/\overline{n}$

Пояснення: n_k – довжина k-ї комбінації нерівномірного коду;

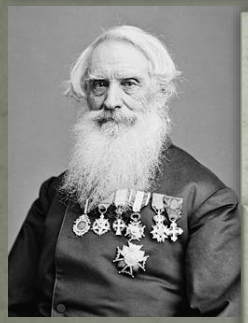
$T_{зн}$ – тривалість видачі знака джерелом;

m – обсяг (основа) коду;

$H(A)$ – ентропія джерела.

Кафедра КІ та кібербезпеки

Першим ефективним статистичним кодом був код Морзе (1837 р.), але він не префіксний, тому його коефіцієнт ефективності незначний.



Семюе́л Фі́нлі Бріз Мо́рзе (*Samuel Finley Breeze Morse*) (1791 – 1872)

Перше повідомлення було послане 24 травня 1844 р. , його текст говорив: «Дивовижні справи твої, Господи!»

«Save our souls» (рятуйте наші душі) був обраний через простоту передачі. Він є комбінацією «. . . — — — . . .»

Першим ефективним префіксним кодом став запропонований ***код Шеннона-Фано*** (1951 р.). Алгоритм побудови коду Шеннона-Фано має таку послідовність процедур за кроками:

1. Упорядкування шляхом розміщення знаків джерела в порядку убування їхніх імовірностей.
2. Поділ знаків на 2 групи з приблизно рівними ймовірностями. Ця процедура призводить до різних кінцевих результатів, залежно від поділу знаків на групи.
3. Знакам верхньої групи (в таблиці) приписується символ „0“, знакам нижньої групи – символ „1“ (або навпаки, це не має істотного значення).
4. Повторення кроків 2 і 3 – доки поділ знаків на групи не закінчиться і їм не будуть приписані символи „0“ чи „1“.
5. Кодова комбінація знака формується шляхом виписування символів, приписаних цьому знаку, зліва направо.

Приклад 3.1. Алфавіт джерела має обсяг $M_A = 6$, знаки в повідомленнях незалежні і мають такі ймовірності: $P(a_1) = 0{,}4$; $P(a_2) = 0{,}2$; $P(a_3) = P(a_4) = 0{,}15$; $P(a_5) = P(a_6) = 0{,}05$, а тривалість видачі одного знака 1 мс. Побудувати код Шеннона-Фано. Обчислити ентропію джерела, середню довжину кодових комбінацій, швидкість цифрового сигналу на виході кодера, порівняти коефіцієнти надмірності на вході та виході кодера, обчислити коефіцієнт ефективності та коефіцієнт стиснення отриманого коду.

Розв'язання. З числа можливих варіантів побудови коду розглянемо алгоритм, показаний в табл. 3.2.

Таблиця 3.2 – Приклад побудови коду Шеннона-Фано

<table>
<tr><th>a_i</th><th>$P(a_i)$</th><th colspan="4">Поділ на групи</th><th>Кодові комбінації</th><th>n_k</th></tr>
<tr><td>a_1</td><td>0,4</td><td>0</td><td colspan="4"></td><td>0</td><td>1</td></tr>
<tr><td>a_2</td><td>0,2</td><td rowspan="5">1</td><td rowspan="2">0</td><td colspan="2">0</td><td>100</td><td>3</td></tr>
<tr><td>a_3</td><td>0,15</td><td colspan="2">1</td><td>101</td><td>3</td></tr>
<tr><td>a_4</td><td>0,15</td><td rowspan="3">1</td><td colspan="2">0</td><td>110</td><td>3</td></tr>
<tr><td>a_5</td><td>0,05</td><td rowspan="2">1</td><td>0</td><td>1110</td><td>4</td></tr>
<tr><td>a_6</td><td>0,05</td><td>1</td><td>1111</td><td>4</td></tr>
</table>

Ентропія джерела

$$H(A) = -\sum_{i=1}^{M_A} P(a_i)\cdot\log_2 P(a_i) = -\left(0{,}4\cdot\log_2 0{,}4 + 0{,}2\cdot\log_2 0{,}2 + 2\cdot 0{,}15\cdot\log_2 0{,}15 + 2\cdot 0{,}05\cdot\log_2 0{,}05\right) =$$

$$= 2{,}25 \text{ дв. од.}$$

Середня довжина кодових комбінацій:

$$\overline{n} = \sum_{k=1}^{M_A} n_k\cdot P(a_k) = 1\cdot 0{,}4 + 3\cdot(0{,}2 + 0{,}15 + 0{,}15) + 4\cdot 2\cdot 0{,}05 = 2{,}3 \text{ дв. симв.}$$

Швидкість цифрового сигналу на виході кодера

$$R = \overline{n}/\overline{T} = 2{,}3/10^{-3} = 2300 \text{ дв.симв./с } (\text{біт/с}).$$

ЮО Бродський Кадрова КІТ Кібербезпеки

Коефіцієнт надмірності джерела *A*

$$K_{\text{надм}}(A) = 1 - \frac{2{,}25}{\log_2 6} = 0{,}13 .$$

Для розрахунку коефіцієнта надмірності джерела *B* (по виходу кодера) необхідно обчислити ентропію цього джерела $H(B)$. Знайдемо ймовірності символів „1“ і „0“ на виході джерела *B*.

$$P(1) = 0{,}2 \cdot \frac{1}{3} + 2 \cdot 0{,}15 \cdot \frac{2}{3} + 0{,}05 \cdot \frac{3}{4} + 0{,}05 = 0{,}354 ;$$

$$P(0) = 0{,}4 + 0{,}2 \cdot \frac{2}{3} + 2 \cdot 0{,}15 \cdot \frac{1}{3} + 0{,}05 \cdot \frac{1}{4} = 0{,}646 .$$

Ентропія джерела *B*

$$H(B) = -\left(0{,}354 \cdot \log_2 0{,}354 + 0{,}646 \cdot \log_2 0{,}646\right) = 0{,}94 \text{ дв. од.}$$

Коефіцієнт надмірності джерела *B*

$$K_{\text{надм}}(B) = 1 - 0{,}94 = 0{,}06 .$$

Отримані значення показують, що кодування зменшує надмірність.

Для порівняння отриманого коду з іншими кодами обчислимо:

коефіцієнт ефективності коду $\mu = \frac{H(A)}{\overline{n}} = \frac{2{,}25}{2{,}3} = 0{,}98 ;$

коефіцієнт стиснення нерівномірного коду $\eta = \frac{n}{\overline{n}} = \frac{3}{2{,}3} = 1{,}3 ,$

де n – довжина рівномірного коду, яка визначається як ціле число, що задовольняє умові $n \geq \log_2 M_A$.

Ю. Бродський Кафедра безпеки

Розглянута методика Шеннона-Фано не завжди приводить до однозначної побудови коду. При розбитті на підгрупи можна зробити більшою за імовірностями як одну, так і іншу підгрупу. Цих недоліків не має методика Хаффмана.

Другим ефективним префіксним кодом є *код Хаффмана* (1952 р.). Цей код є оптимальним префіксним кодом для дискретних джерел без пам'яті за критерієм – мінімальна середня довжина кодової комбінації знака.

Код Хаффмана відіграє важливу роль у кодуванні зображень. Він є основною частиною алгоритму кодування за стандартами JPEG, MPEG і H.261.

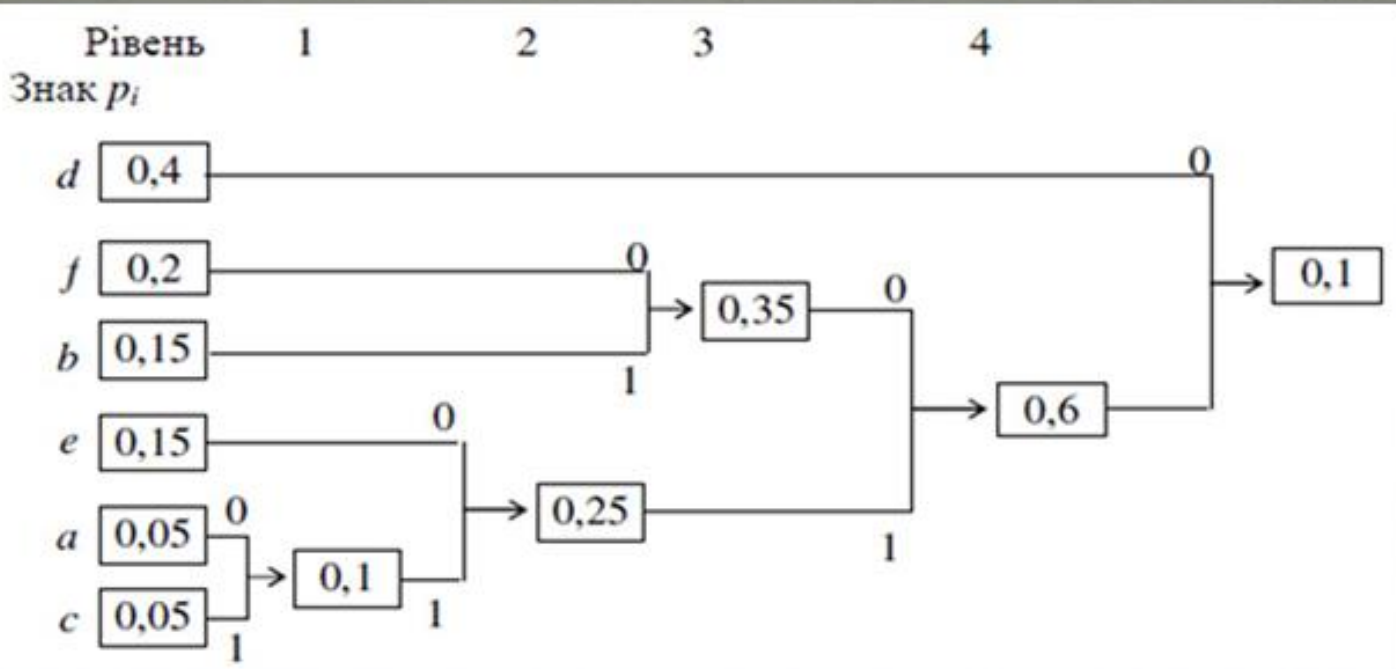
Є стандартним для кодування факсимільних повідомлень згідно з Рекомендацією Е.452 Міжнародного союзу електрозв'язку. Крім того, код Хаффмана використовуються для кодування аудіо сигналів.

АЛГОРИТМ ПОБУДОВИ КОДУ ХАФФМАНА

Він базується на кодовому дереві і має таку послідовність процедур:

1. Упорядкування знаків шляхом розміщення їх у порядку спадання їхніх ймовірностей.
2. Вибирають два вузли (знаки) з найменшими ймовірностями. З них будують дві вітки, які сходяться в один вузол, що відповідає складеному знаку, а його ймовірність дорівнює сумі ймовірностей вузлів, з яких вийшли вітки. Віткам приписують символи 1 і 0, наприклад, верхній вітці 1, а нижній вітці 0.
3. Повторення кроків 1 і 2 – доки не буде досягнуто кореня кодового дерева.
4. Кодова комбінація знака формується шляхом виписування символів, починаючи з кореня кодового дерева, проходячи по вітках до цього знака, тобто справа наліво.

x_i	a	b	c	d	e	f
p_i	0,05	0,15	0,05	0,4	0,15	0,2
$\log_2 p_i$	4,32	2,74	4,32	1,32	2,74	2,32
$H(X)$	$\approx 2{,}25$					



x_i	d	f	b	e	a	c
p_i	0,4	0,2	0,15	0,15	0,05	0,05
Кодове слово	0	100	101	110	1110	1111
Довжина кодового слова	1	3	3	3	4	4

При всій простоті побудови і декодування, коди Хаффмана мають три недоліки:

- Різні довжини кодових слів призводять до нерівномірних затримок декодування.
- Стиснення даних знижує надмірність і тому підвищує схильність до поширення помилок. У випадку кодування Хаффмана це означає, що один, помилково розпізнаний символ, може призвести до того, що цей і всі наступні знаки будуть декодовані невірно.
- Кодування Хаффмана припускає знання ймовірностей подій (знаків), чи, принаймні, деяких оцінок цих ймовірностей. На практиці дуже часто імовірності подій невідомі, а їхні оцінки дуже ускладнені. Саме тому для стиснення великих масивів даних часто використовують універсальний алгоритм кодування, відомий як алгоритм Лемпеля-Зіва.

Ідея **словникових** алгоритмів була запропонована Я. Зівом і А. Лемпелем в 1977 р.

Ідея була втілена в алгоритмі LZ77, що став основою цілого сімейства алгоритмів. Альтернативний підхід, запропонований ними ж в 1978 р., породив сімейство алгоритмів LZ78.

Словникові алгоритми мають не найвищі коефіцієнти ущільнення, але допускають дуже швидку реалізацію, особливо при декодуванні даних, що і зумовлює їх широке використання.

Евристичні (словникові) алгоритми ущільнення (типу LZ77, LZ78), як правило, шукають в файлі рядки, що повторюються, і будують словник фраз, що вже зустрічались. Зазвичай такі алгоритми мають цілий ряд специфічних параметрів (розмір буфера, максимальна довжина фрази і т.п.), підбір яких залежить від досвіду автора роботи, і ці параметри добираються таким чином, щоб досягти оптимального співвідношення часу роботи алгоритму, коефіцієнта стиснення та переліку файлів, що добре стискаються. В наш час алгоритми сімейства LZ77 широко використовуються в програмах універсальних архіваторів. Це, як правило, модифікації LZSS з використанням кодів Хаффмана (LHARC) Шеннона-Фано (PKZIP) або арифметичного кодування (HA).

У алгоритмах сімейства LZ78 ланцюжки в словнику збільшуються на один символ кожного разу, коли ланцюжок із словника зустрічається у вхідному потоці. У алгоритмі LZ78, так само, як і в LZ77, літеральні події і елементи словника чергувалися.

У 1984 р. Т. Велч запропонував алгоритм LZW (за початковими літерами прізвищ Лемпел (Lempel) і Зів (Ziv) (його авторів) і Уелч (Welch) у якому всі літеральні події є елементами словника. Таким чином, у вихідний потік поміщаються тільки посилання на елементи словника, що спрощує алгоритм. Велч показав, що алгоритм може бути легко реалізований апаратно. Програмна реалізація, як правило, виконується на основі вектора двійкових дерев. Корінням дерев служать елементи вхідного алфавіту.

[Майданюк В. П. Методи і засоби комп’ютерних інформаційних технологій. Кодування зображень. Навчальний посібник. – Вінниця: ВДТУ, 2001. – 63 с.]

Часто стиснення – це лише один з етапів загального процесу обробки даних у системах телекомунікацій. Методи стиснення також широко використовують універсальні архіватори, системні утиліти для роботи з жорсткими та компакт-дисками, текстові редактори, програми для роботи зі звуковими, відео- та графічними даними тощо.

Для прикладу розглянемо один з найпростіших методів стиснення даних – метод кодування довжин серій (RLE – run length encoding), який ще називають методом кодування груп. Це адаптивний алгоритм стиску, який зменшує фізичний розмір груп однакових символів.

Групи однакових символів кодують 3-ма байтами – <pref, numb, symb>, де pref – унікальний код, який визначає початок групи, numb – кількість символів у групі, symb – код символу. Наприклад, кодом повідомлення "abccccddddbbcaaaa" буде рядок "ab#4c#4dbbc#4a".

Цей метод буде ефективним за наявності значної кількості чотирисимвольних і більше повторень.

Можливі різні схеми групового кодування:

а) кодування вздовж осі X або Y, коли зображення має безліч горизонтальних або вертикальних ліній;

б) групове кодування, коли зображення складаються з елементів однакових розмірів (виду шахівниці).

Архівація даних

Архівація даних – це каталогізація та стиснення даних з метою їхнього збереження або передавання. Сьогодні найпопулярнішими є програми-архіватори WinRar, WinZip, 7-Zip, PowerArchiver та інші.

WinRar (Євген Рошал) – відомий архіватор з власним форматом RAR. Підтримує метод неперервного стиску, шифрування, створення багатотомних архівів та архівів автономного розпакування (SFX-архівів) тощо. Окрім формату RAR, повністю підтримує формат ZIP, може розпаковувати багато відомих сьогодні архівів.

WinZip ("Corel Corporation") – має зручний інтерфейс, може автоматично розпізнавати типи файлів, підтримує drag-and-drop та інтегрується в контекстне меню. Підтримує архіви типу ZIP, TAR, UUencoded, XXencoded, BinHex, MIME.

7-Zip (Ігор Павлов) – архіватор з високим ступенем стиску. Підтримує ZIP, 7z, RAR, CAB, GZIP, BZIP2 і TAR архіви. Компресія на 2–10 % більша, ніж у PKZip та WinZip, висока швидкість.

PowerArchiver ("ConeXware Inc") – універсальний потужний архіватор з підтримкою всіх популярних форматів: ZIP, RAR, 7-ZIP, CAB, LHA (LZH) , TAR, TAR, GZ, TAR, BZ2, BH, ARJ, ARC, ACE, ZOO, GZ, BZIP2, XXE, MIME, UUE, XPI, EAR, WAR, REP, JAR, BK, QWK, PK3, а також CD-образів ISO, BIN, IMG та NRG. Підтримує SFX-архіви, шифрування за стандартом AES, інтегрується у контекстне меню тощо.

Різновидом архіваторів є програми резервного копіювання (backup copy) – створення архівів цілих дисків разом з образом операційної системи (ОС), наприклад, Acronis True Image, Hendy Backup та інші. Служби резервного копіювання вбудовані також у сучасні ОС і доступні з панелі керування.

Ю. Бродський
(кафедра КБ та ЗІ)

Принцип роботи архіваторів грунтується на пошуку у файлі „надмірної” інформації та подальшому її кодуванні з метою одержання мінімального об’єму. Найвідомішим методом архівації файлів є стиск послідовностей однакових символів. Наприклад, всередині файлу наявні послідовності байтів, що часто повторюються. Замість того, щоб зберігати кожний байт, фіксується кількість символів, що повторююся, та їхня позиція. Нехай файл, що архівується, займає 15 байт і складається з послідовності символів:

B B B B B L L L L L A A A A A

Заглянемо в таблицю кодування ASCII (American standard code for information interchange)

У шістнадцятковій системі

42 42 42 42 42 4C 4C 4C 4C 4C 41 41 41 41 41

Архіватор може представити цей файл наступним (шістнадцятковим) кодом:

01 05 42 06 05 4C 0A 05 41

Це означає: з першої позиції - п’ять разів повторяється символ „B”;

з позиції шість - п’ять разів повторюється символ „L”;

з позиції одинадцять - п’ять разів повторюється символ „A”.

Для зберігання файлу в такій формі необхідно всього 9 байтів, що на 6 байтів менше вихідного.

Вишуканий метод стиску даних, що використовується в тому чи іншому виді практично будь-яким архіватором, – це оптимальний префіксний код – алгоритм Хаффмана.

Ю. Бродський (кафедра КБ та ЗІ)

Графічні формати для збереження зображення

Всі графічні формати діляться на дві великі групи: растрові та векторні

Файли першої з них містять опис кожної точки зображення (представлені у вигляді матриці (bitmap), що складається з маленьких квадратиків різного кольору – пікселів). Найрозповсюдженішими форматами цієї групи є GIF і JPEG, що використовуються в Інтернеті, а також BMP (стандартний формат Windows) та TIFF, який використовується при зберіганні відсканованих зображень а також в поліграфії.

Файли векторних форматів містять не растрові точки, як у фотознімка, а математичні формули, що описують координати ліній. Наприклад, пряма представлена координатами двох точок, а коло – координатами центру та радіуса, тому досягається дуже малий розмір файлу. Векторні графічні формати використовують для передачі схем і рисунків, що складаються з набору ліній, описаних математичними формулами. У векторних форматах зберігаються логотипи, схематичні рисунки, текст, що призначений для виводу на друк та інші подібні об’єкти.

Стандартними засобами перетворити фотозображення у векторний формат практично неможливо, оскільки для цього потрібні спеціальні програми – конвертори. Найрозповсюдженішими розширеннями векторних файлів є: AI для пакета Adobe Illustrator; CDR для пакета CorelDRAW і WMF (ще один «стандартний» формат Windows).

[Основи теорії інформації та кодування : навч. посібник / [І. А. Прокопишин, Р. Є. Рикалюк, В. Ф. Чекурін, К. А. Червінка]. – Електрон. вид. – Львів : ЛНУ ім. Івана Франка, 2023. – 156 с.] URL: https://surl.li/jrhbxk

Висновки:

Таким чином, основними цілями кодування є:

- пристосування форми повідомлення до характеристик каналу зв’язку;
- одержання необхідної множини повідомлень при обмеженій кількості використовуваних елементів;
- досягнення максимальної ефективності системи передачі інформації шляхом зниження надлишковості переданих повідомлень;
- забезпечення необхідної вірогідності передачі інформації в каналах з перешкодами.