Міністерство освіти і науки України
ДВНЗ «Ужгородський національний університет»
Факультет інформаційних технологій
Кафедра інформаційних управляючих систем та технологій

СПОСОБИ І ЗАСОБИ ПЕРЕДАЧІ ІНФОРМАЦІЇ В КОМП'ЮТЕРНИХ СИСТЕМАХ

Методичні вказівки
до курсу для магістрів денної та заочної форм навчань
спеціальностей 121 «Інженерія програмного забезпечення»
та 122 «Комп'ютерні науки та інформаційні технології»

Ужгород - 2016

Способи і засоби передачі інформації в комп'ютерних системах: методичні вказівки до курсу для магістрів денної та заочної форм навчань спеціальностей 121 «Інженерія програмного забезпечення» та 122 «Комп'ютерні науки та інформаційні технології»

У методичних вказівках до курсу «Способи і засоби передачі інформації в комп'ютерних системах» докладно розглядається матеріал пов'язаний з теорією інформації, методами стиску, способами передачі інформації, технічними засобами зв'язку при передачі, коди з виявленням і виправленням помилок

Розробники:
Ніколенко В.В., к.ф.-м.н., доцент кафедри інформаційних управляючих систем та технологій ДВНЗ «Ужгородський національний університет»

Міца О.В., к.т.н., доцент, завідувач кафедри інформаційних управляючих систем та технологій ДВНЗ «Ужгородський національний університет»

Пецко В.І., к.т.н., старший викладач кафедри інформаційних управляючих систем та технологій ДВНЗ «Ужгородський національний університет»

Рецензенти:

Завілопуло А.М., д.ф.-м.н., професор, професор кафедри інформаційних управляючих систем та технологій ДВНЗ «Ужгородський національний університет»;

Повхан І.Ф., к.т.н., доцент, доцент кафедри програмного забезпечення систем, декан факультету інформаційних технологій ДВНЗ «Ужгородський національний університет».

Рекомендовано кафедрою інформаційних управляючих систем та технологій

Рекомендовано Вченою радою факультету інформаційних технологій

Зміст

Вступ

Поняття інформації відноситься до основних понять науки і тісно пов'язане з такими поняттями, як «інформаційні ресурси», «інформаційні технології», «інформаційне забезпечення», «інформаційні системи», «інформаційні структури» тощо. Для того, щоб дати найбільш повне визначення інформації, необхідно звернути увагу на те, що в її основі лежить взаємозалежність пари об'єктів – джерела і споживача інформації. Джерелами інформації, насамперед, є природні об'єкти: люди, тварини, рослини, планети та ін. Разом із цим в міру розвитку науки і техніки джерелами інформації стають наукові експерименти, машини, апарати, технологічні процеси. Значний також перелік об'єктів, що є споживачами інформації: люди, тварини, рослини, різноманітні технічні пристрої тощо. Передача даних (обмін даними, цифрова передача, цифровий зв'язок) — фізичне перенесення даних цифрового (бітового) потоку у вигляді сигналів від точки до точки або від точки до множини точок засобами електрозв'язку каналом зв'язку; як правило, для подальшої обробки засобами обчислювальної техніки.

Дисципліна “Способи і засоби передачі інформації в комп’ютерних системах” знайомить студентів з основними способами передачі інформації, технічними засобами зв’язку при передачі, загальними теоретичними відомостями про інформацію, ентропію джерела інформації, передачу інформації із завадами та без завад, різні методи стиску даних та коди з виявленням і виправленням помилок.

1. ІНФОРМАЦІЙНІ ОЦІНКИ ЕЛЕКТРОННИХ СИСТЕМ.

Найчастіше дають таке означення інформації: **інформація** – «сукупність відомостей про всілякі об'єкти, явища, процеси». Більш змістовним є твердження: інформація є вибір одного варіанта (або декількох) із багатьох можливих і рівноправних. Слово «рівноправних» означає, що усі варіанти, із яких робиться вибір, мають щось спільне, тобто належать одній множині. У наукових працях розглянуто три основних варіанти вказаного вибору: якщо вибір підказаний (або вказаний), то мова йде про одержання (рецепцію) інформації; якщо вибір зроблений самостійно і випадково, то говорять про виникнення інформації; і, нарешті, якщо вибір однозначно визначений ситуацією або попередніми подіями (тобто вибору, по суті, немає), то про інформацію взагалі говорити не доводиться.

Основними *формами подання інформації є символьна, текстова і графічна*.

Символьна форма ґрунтується на використанні символів – літер, цифр, знаків і т. д., є найбільш простою і практично застосовується тільки для передачі сигналів про різні події.

Більш складною є текстова форма подання інформації. Тут, як і в попередній формі, використовуються символи – літери, цифри, математичні знаки. Однак інформація закладена не тільки в цих символах, але й у їх сполученні, порядку проходження. Завдяки взаємозв'язку символів і відображенню мови людини текстова інформація надзвичайно зручна і широко використовується в повсякденному житті.

Найбільш місткою, але і найбільш складною, є графічна форма подання інформації. Образи природи, фотографії, креслення, схеми, малюнки мають велике значення в нашому житті і містять величезну кількість інформації. І хоча інформація не має ні маси, ні геометричних розмірів, жодних фізичних або хімічних властивостей, проте для її існування обов'язкова наявність якогось матеріального об'єкта, що передає або зберігає інформацію. Таких об'єктів досить багато, і їхня кількість увесь час зростає.

Можливі різні класифікації за різними ознаками. Як приклад можна навести одну з класифікацій соціальної інформації, згідно з якою виділяються два основних класи – масова (загальна) і спеціальна (системна) інформація.

Масова інформація – соціальна інформація, адресована всім членам суспільства незалежно від їх становища і роду занять.

Спеціальна інформація адресована не всім членам суспільства, а певним соціальним групам (наприклад, ученим даного фаху, економістам, робітникам якоїсь професії і т. д.). Для сприйняття цієї інформації необхідний початковий запас спеціальних знань і володіння професійною мовою. Назвемо деякі найбільш важливі різновиди спеціальної соціальної інформації.

1. Наукова інформація – утворюється в результаті науково-дослідної діяльності. Наукову інформацію можна визначити як передане в інформаційному процесі наукове знання.

2. Технічна інформація – створюється в сфері техніки і призначена для вирішення технічних задач (розробки нових технічних пристроїв, машин, матеріалів і т. п.). Структура і властивості наукової і технічної інформації дуже близькі, тому ці два види часто об'єднують терміном «науково-технічна інформація».

3. Технологічна інформація – безпосередньо використовується в сфері матеріального виробництва для створення матеріальних благ (продуктів харчування, одягу, машин тощо).

4. Планово-економічна інформація – наприклад, про стан і перспективи розвитку промислових підприємств – використовується для планування і керування виробництвом.

Кожне повідомлення містить як умовну, так і безумовну інформацію. Визначити, яка інформація є умовною і яка безумовною, не так просто, як це здається на перший погляд.

Тут відіграють роль такі обставини:

- по-перше, умовна інформація має тенденцію до уніфікації, що природно, оскільки цінність її при цьому зростає;

- по-друге, уніфікована умовна інформація часто сприймається як безумовна;

- по-третє, найбільш цікавим і гострим залишається питання про умовність (або безумовність) інформації в природничих науках; прийнято вважати, що, вивчаючи природу, ми одержуємо (рецепіюємо) об'єктивну (тобто безумовну) інформацію.

Відомі дві основних форми існування інформації – **статична** (у вигляді записів на папері, стрічці, диску, фотопапері тощо) та **динамічна** – під час її передавання. Потрібно зауважити, що процес фізичного перевезення чи пересування носія інформації (листа, магнітної стрічки, диска, касети тощо) не відноситься до динамічної форми існування інформації. Якщо дані передаються каналом зв’язку, то у кожній точці каналу під час передавання процес змінюється в часі і так само змінюється вплив зовнішніх факторів на сигнали, що несуть в собі інформацію. При фізичному перевезенні цього не відбувається, хоча дані, що зафіксовані на носію, теж підпадають під вплив зовнішніх факторів і можуть руйнуватися з часом. Таким чином, статичною цю форму можна назвати відносно. Більш точне визначення – квазістатична.

Інформація, що зберігається на носію, може зчитуватись, передаватись, знов записуватись, тобто вона може багаторазово переходити з однієї форми існування до іншої.

Основна проблема – передавання інформації з найменшими втратами. При цьому необхідно оцінювати інформацію кількісно.

1.1 Інформація та її характеристики.

Перша спроба ввести науково обґрунтовану міру інформації була зроблена в 1927 році Р. Хартлі (Англія). Він запропонував та обґрунтував кількісну міру, яка дозволяє порівнювати спроможність різних систем передавати інформацію. Ця міра підходить і для систем зберігання інформації, тому вона є відправною точкою для створення теорії інформації.

Природною вимогою, що висувається до інформаційної міри, є вимога адитивності, тобто, кількість інформації, яка може бути збережена у двох однакових комірках, повинна бути удвічі більшою за ту, що зберігається в одній з них.

Якщо одна комірка для зберігання інформації має m можливих станів, то дві таких комірки будуть мати m^2 можливих станів, а n однакових комірок – m^n можливих станів. Це саме стосується і кількості можливих повідомлень. Якщо символ може прийняти значення «0» або «1», то з одного символу можуть бути одержані 2 повідомлення, з двох символів – 4, з трьох – 8 тощо. Таким чином кількість можливих повідомлень визначається кількістю символів, що входять до слова n та кількістю можливих станів символу m: m^n. Тому Р. Хартлі ввів логарифмічну міру інформаційної ємності:

$$C = \log m.$$

Така міра задовольняє вимогу адитивності. Ємність засобу, що складається з n комірок і має m^n станів, дорівнює ємності однієї комірки, помноженої на їх кількість:

$$C_n = \log m^n = n \cdot \log m.$$

За одиницю вимірювання інформаційної ємності вибрана двійкова одиниця – *біт* (binary digit – двійковий знак), що дорівнює ємності однієї комірки з двома можливими станами. Інформаційна ємність C у двійкових одиницях в загальному випадку визначається як:

$$Ca_2 = k_a \log_a m,$$

де k_a – коефіцієнт, що залежить від основи логарифму a.

При використанні для зберігання інформації десяткових комірок більш зручно користуватись десятковими логарифмами. В цьому випадку:

K10 = log 2 10 ≈ 3,32,

тобто, одна десяткова комірка за інформаційною ємністю дорівнює 3,32 двійковим. Одиниця вимірювання кількості інформації в цьому випадку – **діт**.

Якщо від джерела інформації каналом зв'язку передається повідомлення про подію, апріорна імовірність якої на передавальному боці дорівнювала p_1, то після приймання повідомлення апостеріорна імовірність цієї події для приймача інформації дорівнює p_1. Збільшення кількості інформації з урахуванням логарифмічної міри складає:

$$\Delta I = \log\left(\frac{p_2}{p_1}\right) = \log p_2 - \log p_1$$

Для ідеального каналу зв'язку (без завад та спотворень) приймання інформації є вірогідною подією, тобто імовірність p_2 дорівнює одиниці:

$$\Delta I = -\log p_1. \qquad (1.1)$$

Чим меншою буде імовірність p_1, тим більшою буде невизначеність результату, тобто тим більша кількість інформації вміщується у прийнятому повідомленні.

Значення p_1 знаходиться у межах $0 < p < 1$, тобто ΔI завжди додатна величина. Якщо припустити, що може передаватися *na* символів S_a, що відповідають події *A*, n_b символів S_b, що відповідають події *B* тощо, а всього *m* різних символів. Символи S_a, S_b тощо являють собою алфавіт з різних *m* символів. Сума усіх символів *q* складає:

$$q = n_a + n_b + \ldots$$

Згідно з (1.1) приймання символу S_a дає кількість інформації: $\Delta I = -\log p_1$, де p_a – імовірність події *A*.

Тоді у n_a символах міститься кількість інформації $n_a(-\log p_a)$. Загальна кількість інформації складає:

$$I = (n_a \log p_a - n_b \log p_b - \ldots) = \sum_{i=1}^{m} n_i \log p_i. \qquad (1.2)$$

Вираз для визначення середньої кількості інформації, що припадає на один символ, можна отримати, розділивши (1.2) на *q*:

$$I_i = \sum_{i=1}^{m} \frac{n_i}{q} \log p_i \qquad (1.3)$$

У (1.3) відношення $\frac{n_i}{q}$ є апріорною імовірністю появи символу S_a для великих значень n_i та q, $\frac{n_i}{q}$ – імовірність символу S_b тощо.

Тоді:

$$\lim_{q \to \infty} \left(\frac{n_i}{q} \right) = p_i \,.$$

При цьому сума імовірностей: $p_a + p_b + \ldots = 1$`, оскільки одна з усіх m подій *A, B*, ... відбувається обов'язково (повна імовірність подій).

Таким чином, можна отримати вираз для середньої кількості інформації на один символ:

$$I_i = \sum_{i=1}^{m} \frac{n_i}{q} \log p_i \,,$$

де p_i – імовірність i-того символу.

Необхідно підняти вантаж на певний поверх 16-ти поверхового будинку (нумерація поверхів 0-15, N = 16). Скільки біт інформації повністю визначають завдання?

$$I = Log_2 16 = 4 \,.$$

Отже, 4 біти інформації необхідні й достатні для повного зняття невизначеності вибору. У цьому можна переконатися застосуванням логіки вирахування з послідовним розподілом навпіл інтервалів станів. Наприклад, для 9-го поверху:

1) вище 7-го поверху? Так = 1;
2) вище 11-го поверху? Ні = 0;
3) вище 9-го поверху? Ні = 0;
4) вище 8-го поверху? Так = 1.

Підсумок: поверх номер 9 або 1001 у двійковому численні, чотири двійкових розряди.

Якщо в наведеному прикладі на поверхах є по 4 квартири з нумерацією на кожному поверсі 0-3 (M=4), то при адресації вантажу у квартиру буде потрібно ще 2 біти інформації. Такий же результат одержимо, якщо замість незалежної нумерації поверхів і квартир на поверхах (два джерела невизначеності) ми будемо мати тільки наскрізну нумерацію квартир (одне узагальнене джерело):
$I = Log_2 N + Log_2 M = Log_2 16 + Log_2 4 = Log_2 (MN) = Log_2 64 = 6$ тобто, кількість інформації відповідає вимозі адитивності: невизначеність об'єднаного джерела дорівнює сумі невизначеностей вихідних джерел, що відповідає інтуїтивній вимозі до інформації: вона повинна бути однозначною, а її кількість повинна бути тією самою незалежно від способу задання.

1.2 Ентропія джерела інформації

Ентропія являє собою логарифмічну міру безладдя стану джерела повідомлень і характеризує середній ступінь невизначеності стану цього джерела.

Ентропія *H* визначається з формулою згідно з теоремою К. Шеннона, на основі якої, середня кількість інформації, що припадає на один символ, визначається

$$H = \sum_{i=1}^{m} \frac{n_i}{q} \log p_i$$

В інформаційних системах невизначеність знижується за рахунок прийнятої інформації, тому чисельно ентропія *H* дорівнює кількості інформації *I*, тобто є кількісною мірою інформації.

У загальному випадку, відповідно до теорії ймовірностей, джерело інформації однозначно й повно характеризується ансамблем станів $U = \{(u_1),(u_2),...,(u_n)\}$ з ймовірностями станів відповідно $\{p(u_1), p(u_1), p(u_2),..., p(u_n)\}$ за умови, що сума ймовірностей всіх станів дорівнює 1.

Міра кількості інформації, як невизначеності вибору дискретним джерелом стану з ансамблю U, запропонована К. Шенноном в 1946 році й одержала назву ентропії дискретного джерела інформації або ентропії кінцевого ансамблю:

$$H(U) = -\sum_{n=1}^{N} p_n \log_2 p_n$$

Ступінь невизначеності стану об'єкта (або так званого джерела інформації) залежить не тільки від числа його можливих станів, але й від імовірності цих станів. При нерівноймовірних станах варіанти вибору для джерела обмежуються.

Отже, основні властивості ентропії.

1. Ентропія є величиною дійсною й від’ємною, тому що значення ймовірностей p_n перебувають в інтервалі 0-1, значення $\log p_n$ завжди від’ємні, а значення $-p_n \log p_n$ відповідно додатні.

1. Ентропія – величина обмежена, тому що при $p_n \Rightarrow 0$ значення $-p_n \log p_n$ також прямує до нуля,а при $0 < \text{p} < 1$ обмеженість суми всіх доданків очевидна.

3. Ентропія дорівнює 0, якщо ймовірність одного зі станів джерела інформації дорівнює 1, і тим самим стан джерела повністю визначено (імовірності інших станів джерела дорівнюють нулю, тому що сума ймовірностей повинна бути 1).

4. Ентропія джерела із двома станами u_1 й u_2 при зміні співвідношення їхніх імовірностей $p(u_1)=p$ й $p(u_2)=1-p$ визначається виразом

$$H(U)=[p\log p+(1-p)\log p]$$

і досягає максимуму при рівності ймовірностей.

Так, якщо із двох можливих станів імовірність одного з них дорівнює 0,999 то ймовірність іншого стану відповідно дорівнює 1 – 0,999 = 0,001.

5. Ентропія об'єднаних статистично незалежних джерел інформації дорівнює сумі їх ентропій.

6. Ентропія максимальна при рівній імовірності всіх станів джерела інформації:

$$H(U)_{\max}=-\left(\frac{1}{N}\right)\log\frac{1}{N}=\log N\,.$$

В цьому окремому випадку кількісна міра Шеннона збігається з мірою Хартлі. Якщо повідомлення нерівноймовірні, то середня кількість інформації, що вміщується в одному повідомленні, буде меншою.

При використанні двійкової системи з рівними імовірностями виникнення «0» та «1», згідно із формулою Шеннона:

H = – 0,5 · log 2 0,5 – 0,5 · log 2 0,5 = 1.

Ентропія, а разом з нею і кількість інформації, дорівнюють нулю у випадках, коли $p_1 = 0$, або $p_1 = 1$.

Графік ентропії наведено на рисунку 1.1.

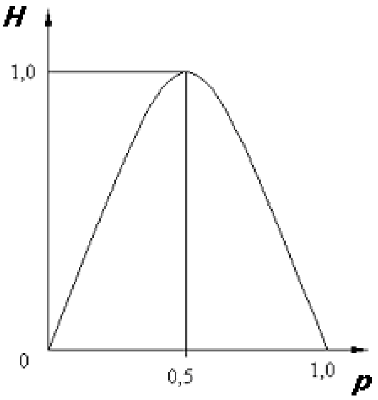

Рисунок 1.1 – Ентропія H для двох можливих станів з імовірностями p1 та (1– p1)

Ентропія, а разом з нею і кількість інформації, дорівнюють нулю у випадках, коли $p_1 = 0$, або $p_1 = 1$.

Для каналів передачі дискретних сигналів (дискретні каналу зв'язку) використають поняття технічної й інформаційної швидкості передачі даних.

Під *технічною швидкістю передачі* (англ. *technical speed of transfer*) розуміють число елементарних сигналів (символів), переданих по каналу за одиницю часу. Найпростіший елементарний символ – однополярний електричний імпульс тривалістю *T*. У дискретних каналах використовують, як правило, двополярні імпульси, позитивні на першій половині інтервалу *T* і негативні на другій половині. Це дозволяє підтримувати нульовий потенціал кабелю й виконувати тактову синхронізацію прийому-передачі сигналів. Одиницею міри технічної швидкості $V_t=1/T$ служить БОД – один символ у секунду. Смуга пропускання каналу зв'язку обмежується граничною частотою $F_{гр}$ за рівнем загасання сигналу до рівня статистичних перешкод, при цьому значення технічної швидкості передачі даних не може бути вище $F_{гр}$ без спеціальних пристроїв виділення інформаційних сигналів.

При відомій технічній швидкості *Vt* швидкість передачі інформації вимірюється в бітах у секунду й задається співвідношенням:

$$V = V_t H(s),$$

де *H(s)* – ентропія символу.

Для двійкових дискретних символів [0, 1] при постійній амплітуді імпульсів значення *H(s)* дорівнює 1. При числі *L* можливих рівноймовірних рівнів амплітуди імпульсів (рівень завад менший різниці рівнів амплітуд імпульсів) значення *H(s)* дорівнює *log L*.

Інформаційна ємність сигналу або повна кількість інформації в сигналі *S* (повідомленні, кодовій послідовності/слові) визначається повною кількістю $N = t/T$ ентропії символів у бітах на інтервалі задання сигналу *t*:

$$I(s) = NLogL = (t/T)LogL.$$

Збільшення числа рівнів *L* збільшує пропускну здатність каналів зв'язку, але ускладнює апаратуру кодування даних і знижує завадостійкість.

Для неперервних сигналів передача по каналах зв'язку можлива тільки за умови, що максимальна інформаційна частота в сигналі F_{max} не перевищує граничної частоти $F_{гр}$ передачі сигналів каналом зв'язку. Для оцінки інформаційної ємності безперервного сигналу виконаємо його дискретизацію з інтервалом $t = 1/2F_{max}$. Як установлено Котельниковим і Шенноном, за миттєвими відліками неперервного сигналу з таким інтервалом дискретизації аналоговий сигнал може бути відновлений без втрати інформації.

При повній тривалості сигналу T_s число відліків:

$$N = t/T = 2F_{\max}T_s.$$

Визначимо максимально можливе число вибірок у кожному відліку при наявності шуму в каналі із середньою потужністю $P_ш=1$. При середній потужності сигналу $P_s=s^2$:

$$L=\sqrt{\frac{s^2+\delta^2}{\delta^2}}=\sqrt{1+\frac{s^2}{\delta^2}}$$

Інформаційна ємність сигналу:

$$I(s)=2F_{ma}T_s LogL\,.$$

Інформаційні можливості сигналу зростають із розширенням його спектра й перевищенням його рівня над рівнем завад.

1.3 Передача інформації без завад

Ємність каналу (англ. *channel capacity*) – гранична швидкість передавання інформації цим каналом:

$$C=\lim_{T\to\infty}\left(\frac{\log q}{T}\right),$$

де q – кількість елементарних інформативних повідомлень, що передається за час T.

Якщо сигнали передаються зі швидкістю S імпульсів за секунду, тобто:

$$S=\frac{1}{\tau},$$

де τ – час передавання одного імпульсу;

то за час T можна передати n імпульсів:

$$n=\frac{T}{\tau}=ST$$

Для двійкового каналу, що пропускає лише елементарні сигнали «0» та «1», максимальна кількість комбінацій елементарних сигналів, яка може бути передана за час T, складає:

$$q=2^n=2^{sr}\,.$$

Тоді ємність бінарного каналу зв’язку визначається:

$$C=\lim_{T\to\infty}\left(\frac{\log_2 q}{T}\right)=\frac{\log_2 2^{sr}}{T}=S\,,$$

тобто, чим меншою буде тривалість імпульсу $\tau = 1/S$, тим більшою буде ємність каналу C. Для недвійкового каналу:

$$q=m^{sr}\,,$$

де m – кількість символів у алфавіті;
і ємність каналу:

$$C=\lim_{T\to\infty}\left(\frac{\log q}{T}\right)=\frac{\log m^{sr}}{T}=S\log m\,.$$

Ємність каналу зв'язку *C* може бути виражена у бітах на символ. Якщо до входу каналу підключене джерело повідомлень з ентропією на символ, що дорівнює ємності каналу зв'язку, то джерело інформаційно узгоджене з каналом. Якщо ентропія джерела менша ніж ємність каналу, то ємність каналу використовується не повністю (канал інформаційно недовантажений).

Узгодження джерела з каналом є досить складною справою і реалізовується за допомогою статистичного кодування. К. Шеннон показав, що інформаційне узгодження, яке досягається статистичним кодуванням, аналогічне енергетичному узгодженню внутрішнього опору електричного генератора з навантаженням за допомогою трансформатора для передавання від генератора максимальної потужності. Тут маємо на увазі узгодження джерела з каналом зв'язку за допомогою пристрою кодування з метою максимального використання ємності каналу.

1.4 Передача інформації із завадами

Завади (англ. *interference*) або *шуми* (англ. *noise*) у каналі зв'язку суттєво ускладнюють передавання інформації. На приймальному боці немає впевненості, що той чи інший елемент повідомлення прийняті у тому вигляді, в якому вони були передані. Тому під час передавання каналом із завадами виникають дві проблеми:
– підвищення ефективності передавання;
– підвищення вірогідності (завадозахищеності) передавання .

Ці проблеми до певної міри протилежні.

Якщо за рахунок впливу шуму був прийнятий елемент повідомлення *j*, в той час, як був переданий елемент*i*, то збільшення інформації можна визначити:

$$\Delta I_{ij} = \log \frac{1}{p_i} - \log \frac{1}{p_{j(i)}} = \log \frac{p_{j(i)}}{p_i},$$

де p_i – апріорна імовірність передавання елемента *i*;
$p_j(i)$ – умовна імовірність приймання елемента *j* в той час, як був переданий елемент *i*.

Якщо шуми досить великі $\left(p_j(i) = p_i, \Delta I = \log l = 0\right)$, повідомлення, що приймається, не вміщує інформації і приймання його не змінює початкових знань. За умови відсутності шуму: $p_j(i) = 1$, якщо $i = j$, або $p_j(i) = 0$, якщо $j \neq i$. В цьому випадку:

$$\Delta I = \log \frac{1}{p_i} = -\log p_i .$$

Пропускна здатність каналу з шумами (у двійкових одиницях на символ) дорівнює середньому за всіма i та j значенню приросту інформації:

$$R_c = \sum_{ij} p_i p_j(i) \Delta I_{ij} = H_i - H_j(i) = H_j - H_i(j)$$

де $H_i = \sum_i p_i \log p_i$ – ентропія джерела;

$H_j = \sum_j p_j \log p_j$ – ентропія повідомлень на приймальному боці;

$H_i(j) = \sum_{ij} p_i(j) \log p_i(j)$, $H_j(i) = \sum_{ij} p_j(i) \log p_j(i)$ – умовні ентропії.

Для каналу з шумами швидкість передавання інформації (у бітах за секунду):

$$v = SR_c,$$

де S – кількість символів, що передаються за секунду.

$$R_c = H_i - H_j(i).$$

Тоді:

$$v = S\left(H_i - H_j(i)\right). \qquad (1.4)$$

Приклад. Якщо швидкість передавання інформації каналу складає 1000 біт/с, а дія завад викликає помилку у 1% символів, то за умови, що імовірності передавання «0» та «1» однакові, ентропія є максимальною $H = 1$ (біт/символ). Імовірність того, що під час передавання «0» приймається «1» складає $p_1(0) = 0{,}01$. Відповідно до цього інші умовні імовірності будуть $p_0(0) = 0{,}99$; $p_0(1) = 0{,}01$; $p_1(1) = 0{,}99$. Розраховані ентропії складають: $H_i = 1$, $H_i(j) = H_j(i) = 0{,}081$. Згідно з (1.4) швидкість передавання інформації каналом із шумами:

$v = 1000 \cdot (1 - 0{,}081) = 919$ (біт/с).

Таким чином швидкість передавання інформації під впливом шуму зменшується більш різко, ніж кількість правильно переданих символів.

К. Шеннон доказав, що якщо ентропія джерела інформації не перевищує пропускної здатності каналу, тобто $H \le C$, то існує код, який забезпечує передавання інформації каналом із шумами з якою завгодно малою частотою помилок або з якою завгодно малою невірогідністю. При $H > C$ такого коду не існує, тобто передавання без помилок неможливе. К. Шеннон визначив також максимальну швидкість передавання інформації:

$$v_k = f_m \log\left(1 + \frac{P_c}{R_ш}\right), \qquad (1.5)$$

де f_m – смуга частот каналу;

P_c – середня потужність сигналу;

$P_ш$ – середня потужність білого шуму.

Таким чином можна передавати інформацію, якщо швидкість передавання інформації не перевищує максимальної швидкості каналу. Для випадку, коли середня потужність білого шуму набагато більша за середню потужність сигналу, тобто $P_c >> P_ш$, у формулі (1.5) одиницею можна знехтувати:

$$v_k = f_m \log(\frac{P_c}{R_ш}).$$

Максимальна кількість інформації, яка може бути передана за певний час T:

$$M_{\max} = f_m T \log(\frac{P_c}{R_ш}). \qquad (1.6)$$

Оскільки ця величина може бути зображена у вигляді паралелепіпеда, то вона отримала назву *об'єму сигналу*. Таким чином можна змінювати окремі параметри сигналу, не змінюючи його об'єм. Якщо до виразу (1.6) підставити потенційні можливості каналу передавання (час, на який канал надається користувачу, виділена йому смуга частот і максимальна потужність сигналу, що може передаватися каналом), то параметр матиме назву *ємність каналу*. Для передавання сигналу каналом зв'язку необхідно щоб об'єм сигналу був не менший, ніж ємність каналу, тобто необхідно виконати умову:

$$V_c \leq V_k. \qquad (1.7)$$

Якщо ж дана умова не виконується, то сигнал передати цим каналом зв'язку неможливо. Може бути ситуація, при якій дана умова виконується, але смуга частот, на яку розрахований канал, менша за смугу частот сигналу або час, який виділено на передавання інформації менший, ніж необхідно, тобто умова (1.7) розпадається на систему:

$$\begin{cases} f_c \leq f_k \\ T_c \leq T_k \\ h_c \leq h_k \end{cases}$$

Одним з найбільш поширених способів перетворення сигналу є варіювання величинами f_c та T_c за їх незмінним добутком. Потужність сигналу, як правило, не збільшується.

Перелік питань для самоконтролю:

1. Що являють собою статична та динамічна форми існування інформації?
1. Що являють собою міри інформації за Хартлі та Шенноном?
3. Які одиниці вимірювання інформації?
4. Що таке ентропія?
5. Що таке ємність каналу передавання?
6. Що таке інформаційне узгодження джерела повідомлення з каналом зв'язку?
7. В чому полягає статистичне кодування? Які його переваги та недоліки?

8. Що розуміють під поняттям “шум”?

9. Чим визначається пропускна здатність каналу зі шумами?

10. Чим визначається швидкість передавання інформації для каналу з шумами?

11. Яка максимальна швидкість передавання інформації?

11. Що таке об’єм сигналу? Як об’єм сигналу пов’язаний з ємністю каналу?

Перетворення сигналів та їх спектральні характеристики

Інформація сприймається споживачем через сигнали, які генеруються або безпосередньо джерелом, або за допомогою генераторів, що не зв’язані з джерелом (локація). Сигнали мають енергетичну та матеріальну основу – світлові потоки, звукові хвилі, теплове випромінювання, зображення точок, неоднорідності електричного і магнітного полів. Сигналами можуть бути документи, книжки, графіки, таблиці і т. ін. Отже, сигнали різняться за фізичною природою. В електронних системах сприйняття сигналів різної фізичної природи здійснюється первинними перетворювачами, які перетворюють інформаційний параметр в сигнал, що є зручним для подальшої обробки.

Важливий елемент інформаційного процесу – передача інформації. Вона відбувається за допомогою повідомлень – послідовності сигналів в часі або просторі, відповідно повідомлення будуть часові або просторові. Повідомлення складаються із матеріальних елементів, якими вони представлені. Це є абетка повідомлення (знаки, символи, колір, частота, інтенсивність і т. ін.), що містить певну кількість елементів. Передача інформації здійснюється за допомогою переносників інформації. Матеріальну основу переносника інформації складає певний фізичний об’єкт або процес, який характеризується набором параметрів. Один із цих параметрів може стати інформаційним, якщо він змінюється під впливом джерела інформації (модуляція параметра). Власне, після модуляції переносник перетворюється в носія інформації.

Як переносники інформації використовуються коливання різної природи, зокрема – гармонічні коливання, включно з нульовою частотою. В електронних системах знайшли поширення переносники у вигляді коливань електричного струму або напруги – електричні сигнали. Якщо частота коливань дорівнює нулю, є лише один інформаційний параметр – рівень струму або напруги. Якщо частота коливань відрізняється від нуля, інформаційними параметрами можуть бути амплітуда, частота або фаза коливань.

Отже, сигнал – змінна фізична величина, що забезпечує передачу інформації лінією зв'язку.

Загалом всі сигнали розподіляються на детерміновані (невипадкові) і випадкові.

Детермінований сигнал характеризується визначеністю його значень в будь-які моменти часу (задаються певною визначеною функцією часу). Випадковий сигнал характеризується тим, що його значення в будь-який момент часу є випадковими величинами. Випадковими можуть бути як корисні (інформаційні) сигнали, так і шкідливі сигнали – завади, які перешкоджають сприйняттю інформаційних сигналів.

Як приклад детермінованого сигналу можна навести синусоїдальну хвилю, яка графічно описується синусоїдою. Синусоїда – це функція часу t, яка записується так:

$$f(t) = A\sin(\omega t + \theta),$$

де величину сигналу визначає коефіцієнт *A*, названий амплітудою;

ω – кутова частота;

θ – початкова фаза.

Синусоїда разом з трикутним, пилкоподібним, прямокутним та іншими сигналами належить до періодичних сигналів, тобто до таких сигналів, які повторюють свою форму через деякий сталий проміжок часу.

Будь-який складний періодичний сигнал може бути поданий за допомогою ряду Фур'є як сума простих гармонічних коливань. Сукупність простих гармонічних коливань, на які може бути розкладений складний періодичний сигнал, називається його спектром.

Розподіл амплітуд гармонік за частотою називають амплітудно-частотним спектром або скорочено амплітудним спектром, а розподіл їхніх початкових фаз за частотою – фазочастотним спектром або фазовим спектром.

Лінії дискретного спектра мають розмірність амплітуди сигналу. Безперервний спектр указує на розподіл амплітуд по всьому спектрі й має розмірність щільності амплітуд сигналу.

Якщо спектр сигналу є необмеженим, то при визначенні ширини нехтують гармоніками, амплітуди яких невеликі й не перевищують певного (заданого) рівня. Найбільш часто користуються рівнем 0,707 за амплітудою або 0,5 за потужністю від максимального значення.

1.5. Перетворення Фур'є

Перетворення Фур'є – інтегральне перетворення однієї комплексно-значної функції дійсної змінної на іншу, тісно пов'язане з перетворенням Лапласа та аналогічне розкладу у ряд Фур'є для неперіодичних функцій. Це перетворення розкладає дану функцію на осциляторні функції . Використовується для того, щоби розрахувати спектр частот для сигналів змінних у часі (таких як мова або електрична напруга). Перетворення названо на честь французького математика Жана Батиста Жозефа Фур'є , який ввів це поняття в 1822 році.

Перетворення Фур'є функції $f(t)$ математично визначається як комплексна функція , яка задається інтегралом

$$F(\omega) = \int_{-\infty}^{\infty} f(t) e^{-i\varpi t} dt .$$

Обернене перетворення Фур'є задається виразом

$$\frac{1}{2\pi} \int_{-\infty}^{\infty} F(\omega) e^{-i\varpi t} dt = f(t).$$

Серед властивостей перетворення Фур'є можна відмітити такі.

Якщо задані інтегровані функції $f(x)$, $g(x)$ та $h(x)$ і їх відповідні перетворення Фур'є $\hat{f}(\xi)$, $\hat{g}(\xi)$ та $\hat{h}(\xi)$ та , тоді перетворення має такі властивості:

а) лінійність:

для довільних комплексних чисел a та b, якщо $h(x) = af(x) + bg(x)$, тоді

$$\bar{h}(x) = a\bar{f}(x) + b\bar{g}(x)$$

б) трансляція:

для довільного дійсного числа x_0, якщо $h(x) = f(x - x_0)$, тоді

$$\bar{h}(\xi) = e^{-2\pi x_0 \xi} \bar{f}(\xi)$$

в) модуляція:

для довільного дійсного числа ξ_0, якщо $h(x) = e^{2\pi x \xi_0} f(x)$ тоді

$$\hat{h}(\xi) = \bar{f}(\xi - \xi_0)$$

г) масштабування:

для нерівного нулю дійсного числа a, якщо $h(x) = f(ax)$, тоді

$$\bar{h}(\xi) = \frac{1}{a} \bar{f}\left(\frac{\xi}{a}\right).$$

Випадок $a = -1$ приводить до властивості обернення часу, згідно з якою: якщо $h(x) = f(-x)$, тоді д) спряження:

Якщо $h(x) = \overline{f(-x)}$, тоді $\bar{h}(\xi) = \overline{\bar{f}(-\xi)}$

Зокрема, якщо f дійсне, тоді має місце умова дійсності $\bar{f}(-\xi) = \overline{\bar{f}(\xi)}$

е) згортка.

Якщо $h(x) = (fg)(x)$, тоді $\bar{h}(\xi) = \bar{f}(\xi) \cdot \bar{g}(\xi)$

Перетворення Фур'є застосовуються для отримання частотного спектру неперіодичної функції, наприклад, електричного сигналу , тобто для подання сигналу у вигляді суми гармонічних коливань . При цьому використовується властивість згортки .

Нехай відгук системи на подразнення у вигляді сигналу $f(t)$буде

$$g(t) = \int_0^\infty \alpha(\tau) f(t-\tau) d\tau ,$$

де $\alpha(\tau)$ – певна функція.

Такий запис означає, що відгук системи залежить не тільки від моментального значення збурення, а також від того подразнення, яке було певний час тому, і яке змінило стан системи.

Застосовуючи перетворення Фур'є до обох частин рівняння, отримуємо

$$G(\omega) = \int_{-\infty}^{\infty} e^{-i\varpi t} \int_0^\infty \alpha(\tau) f(t-\tau) d\tau dt = \frac{1}{2\pi} \int_{-\infty}^{\infty} e^{-i\varpi t} \int_0^\infty \alpha(\tau) \int_{-\infty}^{\infty} F(\omega') e^{i\omega'(t-\tau)} d\omega' dt$$

Оскільки

$$\int_{-\infty}^{\infty} e^{i(\omega'-\omega)e} dt = 2\pi\delta(\omega'-\omega),$$

де $\delta(x)$ – дельта-функція Дірака, інтегрування дає $G(\omega) = A(\omega) F(\omega)$,

де $A(\omega) = \int_0^\infty \alpha(\tau) e^{i\omega\tau}$.

Важливим висновком з цього перетворення є те, що вихідний спектр отримується з вхідного простим множенням на функцію відклику системи $A(\omega)$.

Спектри періодичних і неперіодичних сигналів

Відомо, що будь-яка періодична функція, яка задовольняє умови Діріхле, може бути подана у вигляді нескінченної у загальному випадку суми гармонічних складових – рядом Фур’є. Умова Діріхле полягає у тому, що: функція $x(t)$ повинна бути обмеженою, кусочно-неперервною та мати протягом періоду скінченне число екстремумів.

Відомо дві форми розкладання в ряд Фур'є: тригонометрична й комплексна. Тригонометрична форма розкладання виражається у вигляді

$$x(t) = \frac{1}{2} A_0 + \sum_{k=1}^{\infty} A_k \cos(K\omega_0 t - \varphi_k),$$

де $-\frac{1}{2} A_0$ постійна складова функції $x(t)$;

$\sum_{k=1}^{\infty} A_k \cos(K\omega_0 t - \varphi_k)$ – k-та гармонічна складова;

$A_k, K\omega_0, \varphi_k$ – амплітуда, частота та початкова фаза k-тої гармонічної складової;

$\omega_0 = \frac{2\pi}{T}$ – частота основної (першої) гармоніки;

Т – період зміни функції $x(t)$.

В математичному відношенні зручніше оперувати комплексною формою ряду Фур'є, поданою у вигляді

$$x(t) = \frac{1}{2} \sum_{k=-\infty}^{\infty} A_k \exp(iK\omega_0 t),$$

де $A_k = A_k \exp\{-j\varphi_k\}$ –комплексна амплітуда гармонічної складової частоти $\omega_k = K\omega_0$.

Комплексна амплітуда визначається через тимчасову функцію $x(t)$ за допомогою формули

$$A_k = \frac{2}{T} \int_{t_1}^{t_1+T} x(t) \exp(-jK\omega_0 t) dt.$$

Сукупність амплітуд і відповідних частот гармонік прийнято називати спектром амплітуд.

Сукупність початкових фаз і відповідних частот гармонік називають спектром фаз.

На рис. 1.2 подані графічні зображення спектра амплітуд і спектра фаз періодичного сигналу.

Окремі спектральні складові у графічному зображенні спектра амплітуд називають спектральними лініями.

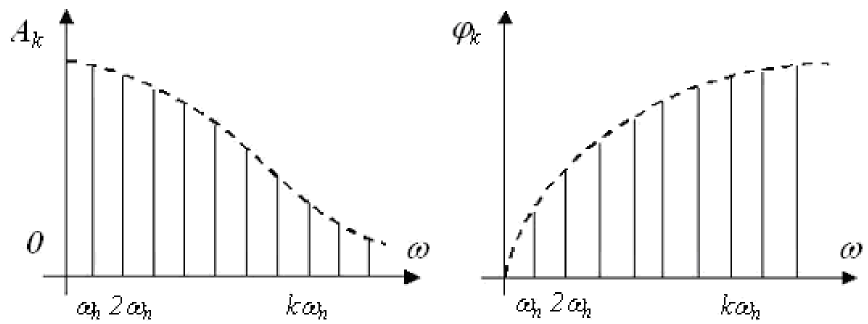

Рисунок 1.2 – Графічні зображення спектра амплітуд і спектра фаз періодичного сигналу.

Будь-який неперіодичний сигнал можна розглядати як періодичний, період зміни якого дорівнює нескінченності. У зв'язку з цим розглянутий раніше спектральний аналіз періодичних процесів може бути узагальнений і на неперіодичний сигнал.

Розглянемо як буде змінюватись спектр неперіодичного сигналу при необмеженому збільшенні періоду зміни сигналу. При збільшенні періоду T інтервали між суміжними частотами в спектрі сигналу і амплітуди спектральних складових зменшується і в границі при $T \to \infty$ стають нескінченно малими величинами. При цьому спектральний розклад неперіодичного сигналу відображається рядом Фур'є.

Комплексна форма неперіодичного сигналу має вигляд

$$x(t) = \frac{1}{2\pi} \sum_{-\infty}^{\infty} S(j\omega)^* \exp(i\omega t) d\omega,$$

де $S(j\omega) = S(\omega)\exp(j\phi(\omega))$ – спектральна щільність сигналу;

$S(\omega) = |S(j\omega)|$ – амплітудно-частотна характеристика сигналу;

$\phi(\omega)$ – фазочастотна характеристика сигналу.

Попередній вираз називається формулою оберненого перетворення Фур'є.

Подання неперіодичної функції інтегралом Фур'є можливе при виконанні таких умов:

1) функція $x(t)$ задовольняє умову Діріхле;

2) функція $x(t)$ абсолютно інтегрована

$$\int_{-\infty}^{\infty} |x(t)| dt.$$

Таким чином, спектр неперіодичного сигналу, на відміну від спектра періодичного сигналу, є суцільним і являє собою суму нескінченної кількості гармонічних складових із нескінченно малими складовими.

Амплітуди гармонічних складових можуть бути подані у такому вигляді

$$dA = \frac{1}{2\pi} S(j\omega)d\omega,$$

звідки спектральна щільність визначається виразом

$$S(j\omega) = 2\pi \frac{dA}{d\omega}.$$

Спектральна щільність пов'язана з функцією часу через пряме перетворення Фур'є

$$S(j\omega) = \int_{-\infty}^{\infty} x(t)^* \exp(-j\omega t)dt.$$

Спектральна щільність однозначно відображає неперіодичний сигнал і задовольняє умову: $\lim_{\omega\to\infty} S(\omega) = 0$.

Модуль спектральної щільності є парною, а аргумент непарною функцією частоти

$$S(\omega) = S(-\omega).\ \varphi(\omega) = -\varphi(-\omega).$$

Спектри одиничних та періодичних імпульсних послідовностей

Аналіз перехідних процесів в колі при дії на нього складної (негармонічної) ЕРС класичним методом виявляється досить складним. Більш зручними в таких випадках є методи, основані на спектральному поданні зовнішньої ЕРС і принципі суперпозиції. Тому, перш ніж переходити до аналізу таких перехідних процесів, розглянемо спектри деяких найбільш важливих для радіотехніки періодичних та неперіодичних ЕРС.

Будь-яку функцію $f(t)$, задану в інтервалі $t_1 < t < t_2$, і яка періодично повторюється з частотою $\Omega = \frac{2\pi}{T}$, де T – період повторення, та таку, що задовольняє умову Діріхле, можна подати рядом Фур'є, який можна записати або в тригонометричній формі

$$f(t) = \frac{C_0}{2} + \sum_{K=1}^{\infty} C(K\Omega)\cos[K\Omega t - \varphi(K\Omega)],$$

де $K = 1,2,3\ldots$,

або в комплексній формі

$$f(t) = \frac{1}{2}\sum_{K=-\infty}^{+\infty} \overline{C}(K\Omega)e^{jK\Omega t},$$

де $\overline{C}(K\Omega) = C(K\Omega)e^{-j\varphi(K\Omega)}$ – комплексна амплітуда K-ї гармонічної складової;

$C(K\Omega)$ – модуль цієї величини, чи просто амплітуда;

$\varphi(K\Omega)$ – початкова фаза K-ї гармонічної складової.

Величина $\frac{C_0}{2}$– середнє за період значення функції $f(t)$ чи постійна складова складної ЕРС

$$\frac{C_0}{2} = \frac{1}{T}\int_0^{\tau} f(t)dt .$$

Розглянемо основні величини, що характеризують спектр складної періодичної ЕРС. Комплексна амплітуда, яка входить в ряд Фур'є, визначається як

$$\overline{C}(K\Omega) = \frac{2}{T}\int_{-\frac{T}{2}}^{\frac{T}{2}} f(t)e^{-jK\Omega t}dt .$$

Залежність модуля комплексної амплітуди від частоти зображають у вигляді графіка, який називається амплітудно-частотним спектром (рис. 1.2, а). Тут кожній частоті $K\Omega$ відповідає лінія, величина якої $C(K\Omega)$ вказує амплітуду гармонічної складової.

Залежність початкової фази від частоти $\varphi(K\Omega)$ зображають у вигляді фазочастотного спектра (рис. 3.2, б). Як видно з рис. 1.2, для складної періодичної ЕРС, спектр є лінійчатим чи дискретним. Тут лінії розміщенні по шкалі частот так, що відділені на відстань , яка дорівнює частоті повторення ЕРС. Зберігаючи незмінним амплітудно-частотний спектр, але змінюючи вигляд фазочастотного спектра, ми тим самим змінюємо форму складної періодичної ЕРС, яка зображена цими спектрами. При розгляді спектра складної ЕРС часто обмежуються одним графіком, на якому зображено амплітудно-частотний спектр з вказуванням фаз гармонічних складових.

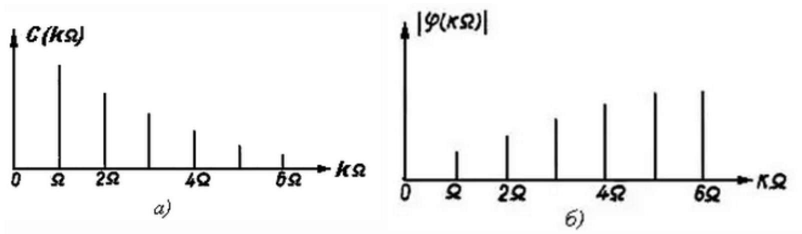

Рисунок 1.3 – Амплітудно-частотний та фазочастотний спектри

Теорема про суму спектрів і теорема запізнення дозволяють обчислити спектр групи однакових рівновідстаючих імпульсів. Нехай є два однакових імпульси $f_1(t)$ і $f_2(t) = f_1(t-\tau)$, розділених інтервалом часу τ. Можна записати спектральну функцію другого імпульсу через спектральну функцію першого:

$$\bar{S}_2(\omega) = \bar{S}_1(\omega) e^{-j\omega\tau}.$$

Тоді на основі виразу для спектральної функції суми двох імпульсів отримаємо:

$$\bar{S}(\omega) = \bar{S}_2(\omega) + \bar{S}_1(\omega) = \bar{S}_1(\omega)(1 + e^{-j\omega\tau}) = \bar{S}_1(\omega)\left[1 + \cos\omega\tau - j\sin\omega\tau\right] =$$
$$= \bar{S}_1(\omega)\sqrt{(1+\cos\omega\tau)^2 + \sin^2\omega\tau}\quad e^{-j\psi(\omega)}$$

де

$$\psi(\omega) = arctg\frac{\sin\omega\tau}{1+\cos\omega\tau}.$$

Зі збільшенням кількості імпульсів спектр групи імпульсів наближається за структурою до лінійчатого спектра періодичної послідовності імпульсів.

Практично всі канали зв’язку мають обмежену смугу пропускання. Отже, при передачі сигналу через реальний канал зв’язку може бути передана лише частина його частотного спектра.

За практичну ширину спектра сигналу приймають діапазон частот, в межах якого знаходиться найбільш вагома частина спектра сигналу. Вибір практичної ширини спектра сигналу визначається двома критеріями: енергетичним критерієм та критерієм допустимих спотворень форми сигналу.

Розглянемо для прикладу послідовність прямокутних імпульсів тривалістю τ, амплітудою *h*, із періодом проходження *T* (рис. 1.4).

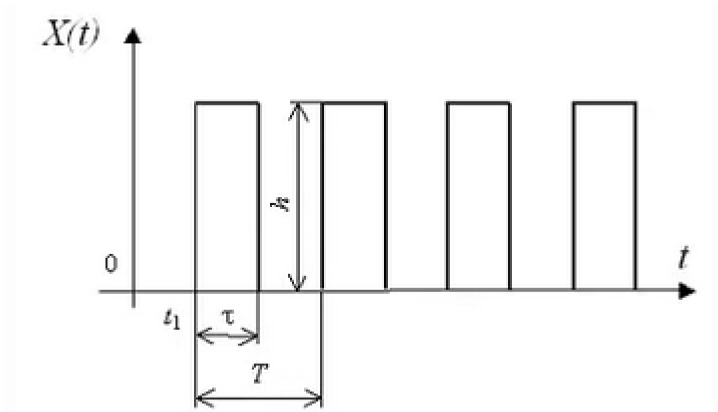

Рисунок 1.4 – Прямокутні імпульси

Розклад в ряд Фур’є періодичної послідовності прямокутних імпульсів подаєтьсяу вигляді

$$x(t) = \frac{\tau h}{T}\left[1 + 2\sum_{k=1}^{\infty}\frac{\sin\frac{K\omega_0\tau}{2}}{\frac{K\omega_0\tau}{2}}\cos K\omega_0\tau\right].$$

Спектр амплітуд такого сигналу показаний на рис. 1.5.

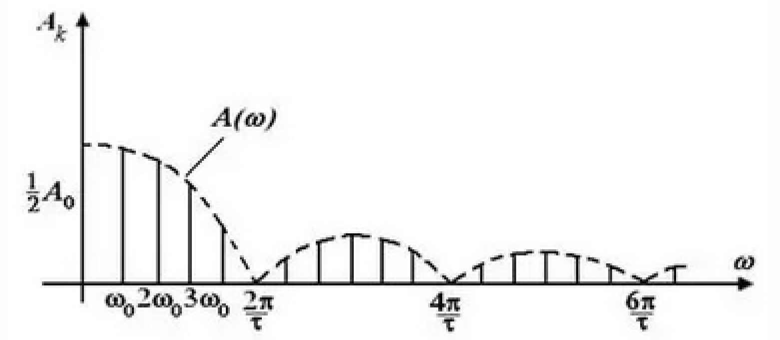

Рисунок 1.5 – Спектр амплітуд

Обвідна його визначається рівнянням

$$A(\omega) = 2\frac{\tau h}{T}\left[\frac{\sin\frac{\omega\tau}{2}}{\frac{\omega\tau}{2}}\right],$$

де $\omega = K\omega_0$ – для K-ої гармоніки.

Можна показати, що для періодичної послідовності імпульсів прямокутної форми тривалістю $\tau = \frac{T}{2}$ достатньо практичну ширину спектра вибрати рівною $3\omega_0 = \frac{6\pi}{T} = \frac{3\pi}{\tau}$. В цій області частот зосереджено 95% всієї потужності сигналу. Розглянемо одиничний прямокутний імпульс тривалістю T та величиною h, спектральна щільність такого сигналу визначається виразом

$$S(j\omega) = \int_{-\infty}^{\infty} x(t)\exp\{-j\omega t\}dt = \int_{-\frac{\tau}{2}}^{\frac{\tau}{2}} h\exp\{-j\omega t\}dt = \tau h\frac{\sin\frac{\omega\tau}{2}}{\frac{\omega\tau}{2}}.$$

Енергія сигналу, зосереджена в смузі частот від 0 до ω_0,

$$\mathrm{W}_1 = \frac{1}{\pi}\int_0^{\omega_0}[S(\omega)]^2 d\omega = \frac{\tau^2 h^2}{\pi}\int_0^1\left[\frac{\sin\frac{\omega\tau}{2}}{\frac{\omega\tau}{2}}\right]^2 d\omega.$$

Для оцінювання впливу ширини смуги пропускання каналу зв'язку на викривлення форми сигналів розглянемо проходження прямокутного імпульсу тривалістю T та величиною U через канал зв'язку, що являє собою ідеальний фільтр низьких частот. Коефіцієнт передачі цього фільтра виражається залежністю: $K(j\omega) = K\exp\{-j\omega T_0\}$, при цьому в діапазоні частот $0 \le \omega \le \omega_0$ модуль коефіцієнта передачі $K(\omega) = K = const$ і аргументу $\varphi(\omega) = \omega T_0$; поза цим діапазоном $K(\omega) = 0$.

В теорії кіл показано, що вихідний сигнал в цьому випадку може бути поданий в аналітичному вигляді

$$a_{вих}(t) = UK\left\{\int_0^{\omega_b} \frac{\sin\omega(t-T_0)}{\omega}d\omega - \int_0^{\omega_b} \frac{\sin\omega(t-(T_0-\tau))}{\omega}d\omega\right\}.$$

Форми переднього й заднього фронтів імпульсу спотворюються однаково. На рис. 1.6 показана форма переднього фронту вихідного сигналу.

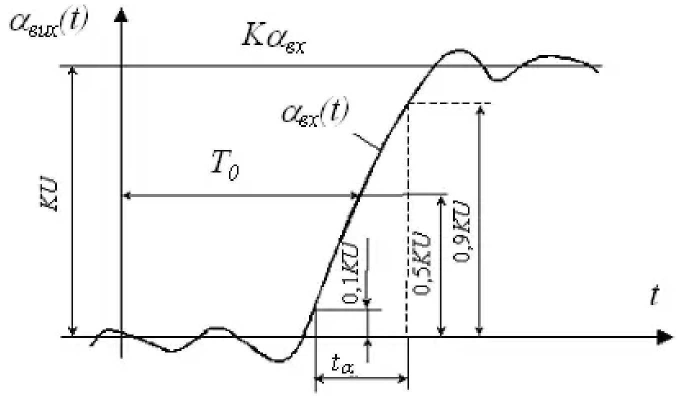

Рисунок 1.6 – Форма переднього фронту вихідного сигналу

Щоб вихідний сигнал зміг досягнути найбільшого значення, активна тривалість переднього фронту $t_{фа}$ повинна бути не більша тривалості вхідного імпульсу τ. При цьому, як показав аналіз, повинна бути справедлива умова $\tau = \dfrac{0,4}{f_b}$ або $f_b \approx \dfrac{0,4}{\tau}$.

Дискретне перетворення Фур'є

Припустимо, що замість функції неперервної змінної $f(x)$ задано функцію дискретної змінної на рівномірній гратці (рис. 3.6), тобто задано значення функції f_k для скінченної послідовності значень аргументу $x_k = kh -$ таблиця функції $\{f_k, kh, k = 0..N\}$ Тут за допомогою $h = \dfrac{L}{N}$ позначено крок гратки – відстань між сусідніми вузлами.

Перетворення Фур'є такої функції можна означити як суму

$$f_k = \frac{1}{\sqrt{N}}\sum_{n=0}^{N-1} f_n e^{-2\pi i\frac{Kn}{N}}$$

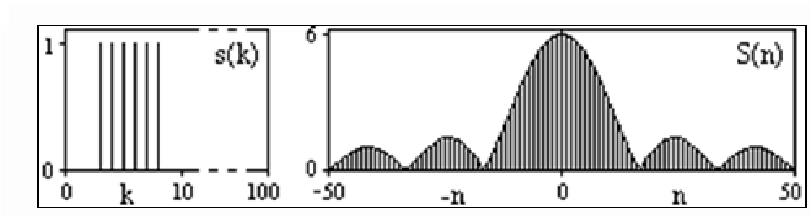

Рисунок 1.7 – Дискретний сигнал та модуль його спектра

Варто зазначити, що недоцільно використовувати суму з кількістю членів, більшою за кількість вузлів гратки.

2. РІЗНІ МЕТОДИ СТИСКУ ІНФОРМАЦІЇ

2.1. Алгоритм Хаффмана

Дані для електронної обробки є деякою послідовністю чисел. Звичайно кожне число в цій послідовності представляється бітовим ланцюжком фіксованої довжини. Наприклад, якщо розглядати дані як послідовність байтів (8-ми бітових шматочків), то бітовий ланцюг[1] 0000000b = 0, 0000001b = 1, 11111111b = 255 і т.д.

У класичній статті, опублікованій в 1952 р., Девід Хаффман описав алгоритм пошуку безлічі кодів, які мінімізують очікувану довжину повідомлень за умови, що відома ймовірність появи кожного символу.

Ідея алгоритму Хаффмана полягає в наступному.

Якщо деякі дані містять різні числа, представлені у вигляді бітових ланцюжків фіксованої довжини і частота[2], з якою різні числа присутні в цих даних, істотно розрізняється, то замінивши бітові ланцюжки фіксованої довжини на бітові ланцюжки різної довжини, причому так, щоб числам, що частіше зустрічаються, відповідали б коротші ланцюжки, можна одержати зменшення об'єму даних.

Далі будемо припускати дані послідовністю байт (8-і бітових шматочків), хоча це необов'язково і можна застосовувати алгоритм Хаффмана до бітових ланцюжків будь-якої фіксованої довжини.

Розглянемо приклад, маємо дані завдовжки в 100 байт, включаючи шість різних чисел: 1, 2, 3, 4, 5, 7.

Стискаючи дані по алгоритму Хаффмана, перше, що необхідно зробити - це підрахувати скільки разів зустрічається кожне число в даних.

Після підрахунку частоти входження кожного числа ми одержуємо таблицю частот (рис. 2.1). Таблиця має ненульові значення частот для чисел 1,

[1] Суфікс «b» означає, що попереднє число записане в двійковому вигляді, а суфікс «.» — число записане в десятковому вигляді.
[2] Кількість разів повторення числа у файлі.

2, 3, 4, 5, 7. Числа з нульовою частотою далі не беруть участь в алгоритмі, про них можна просто забути. Елементи таблиці називатимемо «вузлами».

Таблиця частот (початкова таблиця) методу Хаффмана.

Повна частотна таблиця

Число	0	1	2	3	4	5	6	7	...	254	255
Частота	0	10	20	30	5	25	0	10	...	0	0

Ненульова частина частотної таблиці

Число	1	2	3	4	5	7
Частота	10	20	30	5	25	10

Рис. 2.1

Знайдемо в таблиці найменші частоти. Для цього зручно відсортувати таблицю за збільшенням. У нашому випадку це 5 і 10. Сформуємо з вузлів 5 і 10 новий вузол, частота входження для якого буде рівна сумі частот, рис. 2.2.

Формування першого вузла в методі Хаффмана.

Число	4	1	7	2	5	3
Частота	5	10	10	20	25	30

Новий вузол таблиці

Число	4	1	7	2	5	3
Частота	5	10	10	20	25	30

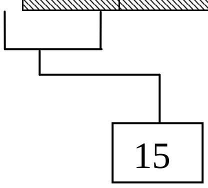

Рис. 2.2

Самі вузли, що утворили новий вузол, більше не беруть участь в створенні нових вузлів.

Новий вузол і вузли таблиці частот, що залишилися, утворюють нову таблицю, для якої повторюють операцію додавання вузла. Найнижча частота 10 (вузол для числа 7) та 15 (новий вузол). Знову додамо вузол, результат зображено на рисунку 2.3.

Формування другого вузла в методі Хаффмана.

Новий вузол таблиці

Число	4	1	7	2	5	3
Частота	5	10	10	20	25	30

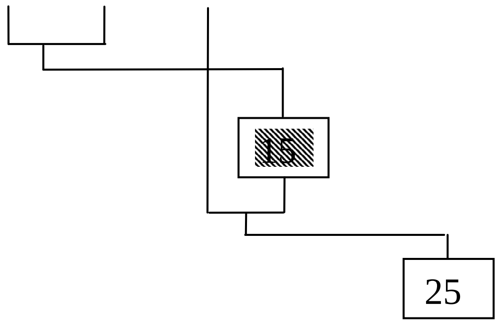

Рис.2.3

Продовжуємо додавати вузли, поки не залишиться єдиний вузол (корінь), рис.2.4.

Дерево вузлів в методі Хаффмана.

Число	4	1	7	2	5	3
Частота	5	10	10	20	25	30

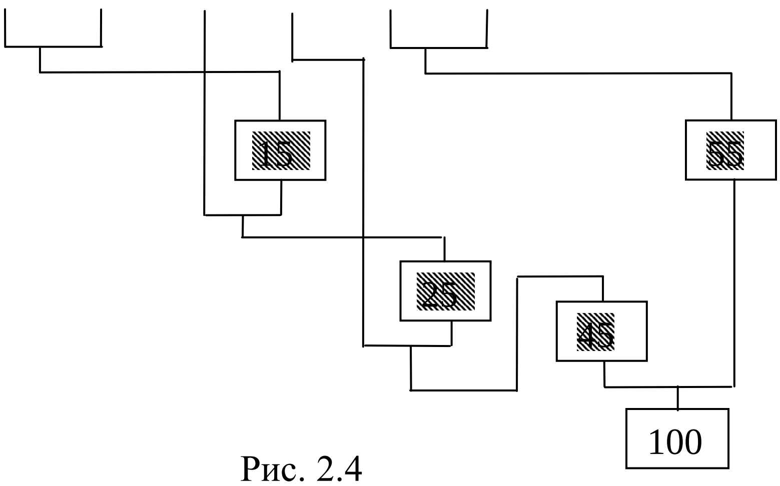

Рис. 2.4

Структура, зображена на рисунку 2.4, де кожен розташований нижче вузол має зв'язок з двома вище розміщеними вузлами носить назву «двійкове дерево» (далі просто дерево).

Тепер коли дерево створене, можна обчислити коди (бітові ланцюжки для кодування початкових чисел) і закодувати дані. Обчислення коду числа починається від кореня дерева. Для обчислення коду, необхідно, рухаючись по дереву від кореня до числа в початковій таблиці, підрахувати число пройдених вузлів. Це значення буде рівне довжині бітового ланцюжка коду. І прослідкувати для кожного вузла повороти, якщо поворот у вузлі здійснюється наліво, то в ланцюжку бітів встановлюється значення 0, якщо направо — 1.

Наприклад, для числа 3 від кореня до початкового вузла пройдемо два вузли, тобто довжина коду два біти. Перший від кореня поворот направо 1 і другий теж направо 1 — код Хаффмана для числа рівний 11b. Виконавши обчислення для всіх чисел одержимо наступні коди Хаффмана:

1. 0111b (4 бита);
2. 00b (2 бита);
3. 11b (2 бита);
4. 0110b (4 бита);
5. 10b (2 бита);
7. 010b (3 бита).

(2.1)

Кожен символ спочатку представлявся 8-ма бітами, і оскільки ми зменшили число бітів необхідних для представлення кожного символу, ми, отже, зменшили розмір вихідного файлу. Результуюче стиснення може бути обчислено таким чином, див. табл. 2.1.

Таблиця 2.1

Підрахунок степені стиску даних.

Число	Частота	Початковий розмір, біт	Ущільнений розмір, біт
I	II	III	IV
1	10	80	40
I	II	II	IV
2	20	160	40
3	30	240	60
4	5	40	20
5	25	200	50
7	10	80	30
Всього	100	800	240

Первинний розмір даних: 100 байт або 800 біт; розмір стислих даних: 30 байт або 240 біт; отже одержали розмір даних 0,3 початкового.

Відмітимо, що в тій же пропорції буде зменшений розмір даних для довшої порції даних, якщо відносні частотні характеристики не зміняться. Слід пам'ятати, що для відновлення початкових даних, необхідно мати декодуюче дерево. Отже, ми повинні зберегти дерево разом з даними. Це приводить у результаті до збільшення розмірів стислих даних.

Якщо для посилання на вузли використовувати номер вузла в масиві вузлів, то такий номер може змінюватися від 0 до 511 (для байта), тобто для зберігання посилання необхідне 9 біт. У розглянутому прикладі дерево складається з 5 вузлів по два посилання — 18 біт, разом 90 біт необхідно для збереження інформації по дереву.

Правильний підрахунок ступеня стиснення буде виглядати так:

первинний розмір даних - 100 байт або 800 біт;

розмір стислих даних - 240 біт і 90 біт даних про дерево.

Так що одержали розмір даних 0,41 початкового. Метод Хаффмана не завжди здатний зменшити розмір даних. Очевидно, що для даних, де зустрічаються всі числа (0.255) і частоти їх рівні вийде довжина коду для кожного числа 8 біт і ніякого стиснення.

Мабуть можна придумати і інші поєднання що приводять до аналогічного результату. Тут слід відмітити, що неможливість стиснути дані по алгоритму Хаффмана не означає відсутність можливості стиснення взагалі.

Наприклад, очевидно, що будь-яка послідовність, де зустрічаються всі числа (0.255) з рівними частотами не стискувана по Хаффману. Але, якщо числа в даних розташовані групами, що складаються з однакових чисел (1,1,1,1,7,7,7,7 і т.д.), то такі дані легко стиснути методом Running.

Адаптивний (динамічний) алгоритм Хаффмана

Описаний вище алгоритм вимагає попередньої обробки всіх початкових даних. Повинні бути обчислені частоти входження для байтів. Після чого дані можуть бути упаковані, тобто необхідно два перегляди початкових даних. Крім того, цей алгоритм вимагає збереження даних про частоти разом з упакованими даними, інакше розпаковування неможливе. Алгоритми стиснення, які володіють такими властивостями, називають статичними.

Ці алгоритми важко використовувати при кодуванні потоку даних, наприклад, при передачі даних по мережі. Існують алгоритми стиснення, які не вимагають попередньої обробки всього масиву даних, а здатні кодувати (упаковувати) дані у міру їх надходження (потік даних). Цим алгоритмам досить одноразового перегляду даних. Алгоритми цього типа називають адаптивними (динамічними).

Хоча метод Хаффмана за своєю суттю статичний, але існує можливість реалізувати адаптивний варіант цього методу.

Псевдокод алгоритму цього методу зображений нижче:

```
void Encode() // Кодування
    Ініціалізація дерева
    while Не_кінець_файла do
      Зчитування символу
```

```
    Кодування символу
    Перестроювання дерева
  end
end
void Decode() // Декодування
  Ініціалізація дерева;
  while Умова завершення вихідного файлу do
    Декодування символу
    Перестроювання дерева
  end
    end
```

Динамічне стиснення методом Хаффмана було запропоновано незалежно Фоллером [1973] і Галлагером [1978]. У 1985 Дональд Кнут розробив остаточний вдосконалений варіант алгоритму – в наслідок чого алгоритм одержав назву FGK (Faller, Gallager, Knuth) алгоритму. Основним принципом FGK алгоритму - є властивість "братерства". Двійкове дерево володіє даною властивістю, якщо кожен його вузол (за винятком кореневого) має в дереві вузла "брата" і, при розташуванні всіх вузлів в списку, у порядку зменшення їх ваги, вузли "брати" знаходяться по сусідству. "Братами" назвемо два вузли, які мають загального батька. Приклад такого дерева зображений на рисунку 2.6.

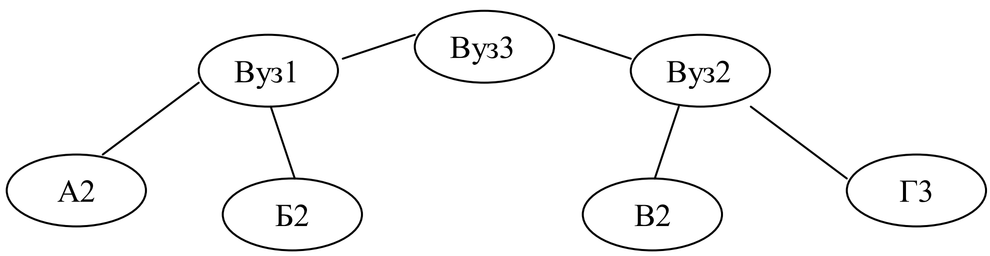

Рис.2.5

Для вказаного дерева, виписавши вузли в потрібний список -отримаємо:

- Вуз3 – 9;
- Вуз2 – 5;

- Вуз1 – 4;
- Г – 3;
- В – 2;
- Б – 2;
- А – 2;

Де верхній ряд означає вузол дерева, а нижня його вага. Таким чином з'ясовуємо, що дане дерево володіє властивістю "братерства", оскільки кожен, окрім кореневого, вузол має "брата" і всі вузли "брати" знаходяться в списку по сусідству { А - Б } { У - Г } { Уз1 - Уз2 }.

Листя такого дерева є символи витягуванні з початкового файлу, а вага листя - кількість появ цих символів в початковому файлі. У міру проходження алгоритму через файл ваги листя можуть або збільшуватися, або залишатися без змін. Спочатку дерево складається з одного "порожнього" вузла. Цей вузол є єдиним листом, який не містить ніякого символу і має нульову вагу. Він призначається для кодування тих символів, які раніше не потрапляли у вхідний потік. Далі, у міру надходження на вхід програми архіватора символів з початкового файлу, перевіряється, чи є вже такий символ в дереві чи ні.

Якщо символ присутній - його код передається стислому файлу, потім його (і всіх його батьків) вага інкрементується і дерево перераховується заново. За відсутності символу в дереві - в стислий файл передається код "порожнього" символу, наступні за яким біти містять інформацію однозначно визначальну новий символ (найчастіше це ASCII код). Після чого в дерево додається ще один лист, який містить новий символ і має вагу 1. Інкрементуються ваги всіх батьків новоспеченого вузла. Знову перераховуємо дерево.

Оскільки "порожній" вузол має вагу 0, то при перерахунку дерева він завжди матиме найдовший код. Отже, кожен новий символ, що з'явився, матиме код рівний по довжині коду "порожнього" вузла або менший його на 1 біт. Дякуючи чому найбільш рідкозустрічаємі символи, які будуть, відповідно, пізніше з'являтися - матимуть щонайдовші коди - що повністю відповідає принципам алгоритму стиснення Хаффмана.

2.2. Алгоритм Шеннона-Фано

Кодування Шеннона-Фано є одним з найперших алгоритмів стиснення, який вперше сформулювали американські учені Шеннон (Shannon) і Фано (Fano). Даний метод стиснення має велику схожість з кодуванням Хаффман, яке з'явилося на декілька років пізніше. Головна ідея цього методу - замінити символи, що часто зустрічаються, коротшими кодами, а послідовності, що рідко зустрічаються, довшими кодами. Таким чином, алгоритм ґрунтується на кодах змінної довжини. Для того, щоб декомпресор згодом зміг розкодувати стислу послідовність, коди Шеннона-Фано повинні володіти унікальністю, тобто, не дивлячись на їх змінну довжину, кожен код унікально визначає один закодований символ і не є префіксом будь-якого іншого коду.

Розглянемо алгоритм обчислення кодів Шеннона-Фано (для наочності візьмемо за приклад послідовність 'aa bbb cccc ddddd'). Для обчислення кодів, необхідно створити таблицю унікальних символів повідомлення c(i) та їх ймовірностей p(c(i)), і відсортувати її у порядку не зростання ймовірності символів, рисунок 2.7.

c(i)	p(c(i))
d	5/17
c	4/17
space	3/17
b	3/17
a	2/17

Рис. 2.7

Далі, таблиця символів ділиться на дві групи так, щоб кожна з груп мала приблизно однакову частоту по сумі символів. Першій групі встановлюється початок коду в '0', другий в '1'. Для обчислення наступних біт кодів символів, дана процедура повторюється рекурсивно для кожної групи, в якій більше одного символу. Таким чином для нашого випадку одержуємо коди символів зображених на рисунку 2.8.

Символ	Код
d	00
c	01
space	10
b	110
a	111

Рис.2.8

Довжина коду s(i) в одержаній таблиці рівна int(-lg p(з(i))), якщо символи вдалось розділити на групи з однаковою частотою, інакше, довжина коду рівна int(-lg p(з(i)))+ 1.

$$\mathrm{int}(-\lg p(c(i))) <= s(i) <= \mathrm{int}(-\lg p(c(i))) + 1 \qquad (2.2)$$

Використовуючи отриману таблицю кодів (рисунок 2.8), кодуємо вхідний потік - замінюємо кожен символ відповідним кодом. Для декодування одержаної послідовності, дану таблицю необхідно зберігати разом із стислим потоком, що є одним з недоліків даного методу. У стислому вигляді, наша послідовність зображена на рисунку 2.9, довжиною в 39 біт.

111111011011011010010101011000000000000

Рис. 2.9

Враховуючи, що оригінал мав довжину рівну 136 біт, одержуємо коефіцієнт стиснення ~28%.

Дивлячись на одержану послідовність, виникає питання: "А як же тепер це розтиснути?".

Ми не можемо, як у разі кодування, замінювати кожні 8 біт вхідного потоку, кодом змінної довжини. При розтисненні нам необхідне все зробити навпаки - замінити код змінної довжини символом завдовжки 8 біт.

В даному випадку, краще всього використовуватиме бінарне дерево, листям якого будуть являтися символи (аналог дерева Хаффмана).

2.3. Арифметичний алгоритм

У 70-і роки у алгоритму Хаффмана з'явився гідний конкурент - арифметичне кодування. Цей метод заснований на ідеї перетворення вхідного потоку в одне число з плаваючою комою. Зрозуміло, що чим довше повідомлення, тим довше число, що виходить в результаті кодування. Отже, на виході арифметичного компресора виходить число, менше 1 і більше або рівніше 0. З цього числа можна однозначно відновити послідовність символів, з яких його було побудовано.

Розглянемо роботу арифметичного компресора на прикладі повідомлення “BILL GATES”.

Поставимо у відповідність кожному символу повідомлення ймовірність його появи в повідомленні (табл. 2.2).

Таблиця 2.2

Ймовірність появи символів у повідомленні

Символ	Ймовірності
I	II
Пробіл	1/10
A	1/10
B	1/10
E	1/10
G	1/10
I	II
I	1/10
L	2/10
S	1/10
T	1/10

Потім привласнимо кожному символу інтервал ймовірності в проміжку від 0 до 1. Довжина інтервалу для символу рівна ймовірності його появи в повідомленні. Положення інтервалу ймовірності кожного символу не має значення. Важливо тільки те, щоб і кодер, і декодер розташовували символи за однаковими правилами. Інтервали ймовірності для символів нашого повідомлення приведені в табл. 2.3.

Таблиця 2.3

Інтервали ймовірностей для символів повідомлень.

Символ	**Ймовірність**	Інтервал
Пробел	1/10	[0.00, 0.10]
A	1/10	[0.10, 0.20]
B	1/10	[0.20, 0.30]
E	1/10	[0.30, 0.40]
G	1/10	[0.40, 0.50]
I	1/10	[0.50, 0.60]
L	2/10	[0.60, 0.80]
S	1/10	[0.80, 0.90]
T	1/10	[0.90, 1.00]

У загальному вигляді алгоритм арифметичного кодування може бути описаний таким чином:

НижняГраниця = 0.0;

ВерхняГраниця = 1.0;

Поки ((НаступнийСимвол =

ДайНаступнийСимвол()) != КІНЕЦЬ)

Інтервал = ВерхняГраниця - НіжняГраниця;

ВерхняГраниця = НіжняГраниця + Інтервал *

ВерхняГраницяІнтервалаДля

(НаступнийСимвол);

НижняГраниця = НижняГраниця + Інтервал *

НижняГраницяІнтервалаДля

(НаступнийСимвол);

Кінець Поки

Видати (НижняГраниця)

Для нашого прикладу цей алгоритм видасть результати (нижні та верхні границі) зображені у таблиці 2.4.

Таблиця 2.4

Кроки алгоритму арифметичного кодування при обробці повідомлення

НаступнийСимвол	НижняГраниця	ВерхняГраниця
	0.0	1.0
B	0.2	0.3
I	0.25	0.26
L	0.256	0.258
L	0.2572	0.2576
Пробел	0.25720	0.25724
G	0.257216	0.257220
A	0.2572164	0.2572168
T	0.25721676	0.2572168
E	0.257216772	0.257216776
S	0.2572167752	0.2572167756

Таким чином, згідно нашій схемі, число 0.2572167752 однозначно кодує повідомлення “BILL GATES”.

Алгоритм арифметичного декодування повинен проробити теж саме, але навпаки. Він може бути описаний ось так:

```
Число = ПрочитатиЧисло();
Завжди
        Символ =
                НайтиСимволВінтервалЯкогоПотрапляєЧисло
                                (Число)
        Видати (Символ) Інтервал =
                ВерхняГраницяІнтервалаДля (Символ) -
                        НижняГраницяІнтервалаДля (Символ);
        Число = Число - НижняГраницяІнтервалаДля(Символ);
        Число = Число / Інтервал; Конец Всегда
```

У приведеному алгоритмі не розглядається питання зупинки декодера. Інтуїтивно можна здогадатись, що його зупинка повинна відбутися після того, як він знайде кінець файлу, який розкодовується. Кінцем файлу буде виступати певний ключ (спеціальний символ, або група символів).

Для нашого прикладу результати роботи декодера можна переглянути у таблиці 2.5.

Таблиця 2.5

Кроки роботи алгоритму арифметичного декодування

Число	Символ	НижняГраниця	ВерхняГраниця	Інтервал
0.2572167752	B	0.2	0.3	0.1
0.572167752	I	0.5	0.6	0.1
0.72167752	L	0.6	0.8	0.2
0.6083876	L	0.6	0.8	0.2
0.041938	Пробіл	0.0	0.1	0.1
0.41938	G	0.4	0.5	0.1
0.1938	A	0.2	0.3	0.1
0.938	T	0.9	1.0	0.1
0.38	E	0.3	0.4	0.1
0.8	S	0.8	0.9	0.1

Таким чином при обробці числа після визначення символу, в інтервал якого воно потрапляє, цей символ видається як розкодований, а його вплив на число усувається діями, зворотними діям при кодуванні.

2.4. Алгоритм Лемпеля – Зіва – Велча

Алгоритм Лемпеля – Зіва – Велча (Lempel – Ziv – Welch, LZW) – універсальний алгоритм стиснення даних без втрат, створений Абрахамом Лемпелем (Abraham Lempel), Якобом Зівом (Jacob Ziv) і Террі Велчем (Terry Wech). Метод був опублікований Велчем в 1984 році, в якості покращеної реалізації алгоритму LZ78, опублікованого раніше Лемпелем і Зівом в 1978 році (звідси й назва алгоритму - LZ78). Алгоритм розроблений так, що його можна швидко реалізувати, але він не обов'язково оптимальний, оскільки не проводить ніякого аналізу вхідних даних.

Опис алгоритму

Акронім «LZW» вказує на прізвища винахідників алгоритму: Лемпель, Зів і Велч, проте багато хто стверджує, що, оскільки патент належав Зіву, то і метод повинен називатися алгоритмом Зіва – Лемпеля – Велча.

Даний алгоритм при стиснені (кодуванні) динамічно створює таблицю перетворення рядків: певним послідовностям символів (словам) ставляться у відповідність групи біт фіксованої довжини (зазвичай 12-бітні). Таблиця ініціалізується всіма 1-символьними рядками (у випадку 8-бітних символів – це 256 записів). По мірі кодування, алгоритм проглядає текст символ за символом, і зберігає кожен новий, унікальний 2-символьний рядок в таблиці у вигляді пари код/символ, де код посилається на відповідний перший символ. Після того як новий 2-символьний рядок збережений у таблиці, на вихід передається код першого символу. Коли на вході читається черговий символ, для нього по таблиці знаходиться рядок, що вже зустрічався, максимальної довжини, після чого в таблиці зберігається код цього рядка з наступним символом на вході; на вихід видається код цього рядка, а наступний символ використовується в якості початку наступного рядка.

Алгоритму декодування на вході потрібен лише закодований текст, оскільки він може відтворювати відповідну таблицю перетворень безпосередньо по закодованому тексту.

Алгоритм

1. Ініціалізація словника всіма можливими односимвольними фразами. Ініціалізація вхідної фрази W першим символом повідомлення.
2. Знайти в словнику рядок W найбільшої довжини, який співпадає з останніми прийнятими символами.
3. Зчитати черговий символ K з повідомлення, що кодується.
4. Якщо КІНЕЦЬ_ПОВІДОМЛЕННЯ, то видати код для W, інакше
5. Якщо фраза WK вже є у словнику, присвоїти вхідній фразі W значення WK та перейти до Кроку 3, інакше видати код W, додати WK у словник, присвоїти вхідній фразі W значення K та перейти до кроку 3.
6. Кінець.

Застосування

На момент своєї появи алгоритм LZW давай кращий коефіцієнт стиснення для більшості застосувань, ніж будь-який інший добре відомий метод того часу. Він став першим широковживаним на комп’ютерах методом стиснення даних.

Алгоритм був реалізований в програмі compress, яка стала в певній мірі стандартною утилітою Unix-систем приблизно в 1986 році. Кілька інших популярних утиліт-архіваторів також використовують цей метод або близькі до нього.

В 1987 році алгоритм став частиною стандарту на формат зображень GIF. Він також може (опціонально) використовуватися у форматі TIFF.

На даний момент реалізація алгоритму міститься у програмі Adobe Acrobat.

Приклад

Даний приклад демонструє алгоритм LZW в дії, зображуючи стан вихідних даних та словника на кожному кроці, як при кодуванні, так і при розкодуванні повідомлення. Задля спрощення викладу, обмежимося простим алфавітом – тільки прописні букви, без знаків пунктуації та пропусків. Повідомлення, яке необхідно стиснути, наведено нижче:

```
TOBEORNOTTOBEORTOBEORNOT#
```

Маркер # використовується для позначення кінця повідомлення. Отже, в алфавіті, що розглядається, 27 символів (26 прописних букв від A до Z і символ #). Комп’ютер записує це у вигляді груп бітів, для запису кожного символу алфавіту достатньо групи з 5 бітів на символ. У процесі збільшення словника, розмір груп має збільшуватися, щоб врахувати нові елементи. 5-бітні групи дають $2^5 = 32$ можливих комбінацій бітів, тому, коли в словнику з’явиться 33-є слово, алгоритм повинен перейти до 6-бітних груп. Зауважимо, що, оскільки використовується група з всіх нулів *00000*, то 33-я група має код 32. Початковий словник міститиме:

```
# = 00000
A = 00001
B = 00010
C = 00011
.
.
.
Z = 11010
```

Кодування

Без використання алгоритму LZW, при передачі повідомлення у початковому вигляді – 25 символів по 5 бітів на кожен символ – воно займає 125 бітів. Порівняємо це з тим, що отримаємо при використанні LZW:

```
Символ:  Бітовий код:  Новий запис у словнику:
         (на виході)

T       20 =  10100
O       15 =  01111     27: TO
B        2 =  00010     28: OB
```

```
E       5 =  00101     29: BE
O      15 =  01111     30: EO
R      18 =  10010     31: OR <--- з наступного символу
N      14 = 001110     32: RN      починаємо використовувати
O      15 = 001111     33: NO      6-бітні групи
T      20 = 010100     34: OT
TO     27 = 011011     35: TT
BE     29 = 011101     36: TOB
OR     31 = 011111     37: BEO
TOB    36 = 100100     38: ORT
EO     30 = 011110     39: TOBE
RN     32 = 100000     40: EOR
OT     34 = 100010     41: RNO
#       0 = 000000     42: OT#

Загальна довжина = 6*5 + 11*6 = 96 бітів.
```

Таким чином, використовуючи алгоритм стиснення LZW ми скоротили повідомлення на 29 бітів з 125 – це майже 22%. Якщо повідомлення буде довшим, то елементи словника будуть представляти все більш і більш довгі частини тексту, завдяки чому слова, що повторюються, будуть представлені дуже компактно.

Декодування

Тепер уявімо, що отримане закодоване повідомлення, наведене вище, потрібно декодувати. Перш за все необхідно знати початковий словник, а подальші записи словника можна реконструювати вже в процесі декодуванні, оскільки вони являються просто конкатенацією попередніх записів.

```
Дані:         На виході:     Новий запис:
                         Повний:       Частковий:
10100  = 20     T                     27: T?
01111  = 15     O        27: TO       28: O?
00010  = 2      B        28: OB       29: B?
00101  = 5      E        29: BE       30: E?
01111  = 15     O        30: EO       31: O?
10010  = 18     R        31: OR       32: R? <--- починаємо
```

```
001110 = 14     N       32: RN      33: N?     використовувати
001111 = 15     O       33: NO      34: O?     6-бітні групи
010100 = 20     T       34: OT      35: T?
011011 = 27     TO      35: TT      36: TO? <--- для 37 додаємо
011101 = 29     BE      36: TOB     37: BE?    тільки перший
011111 = 31     OR      37: BEO     38: OR?    елемент наступного
100100 = 36     TOB     38: ORT     39: TOB?   слова словника
011110 = 30     EO      39: TOBE    40: EO?
100000 = 32     RN      40: EOR     41: RN?
100010 = 34     OT      41: RNO     42: OT?
000000 = 0      #
```

Єдина невелика складність може виникнути, якщо нове слово словника пересилається негайно. У наведеному вище прикладі декодування, коли декодер зустрічає перший символ – **T**, він знає, що слово 27 починається з T, але чим воно закінчується? Проілюструємо проблему наступним прикладом. Ми декодуємо повідомлення **ABABA**:

```
Дані:        На выході:      Новий запис:
                        Повний:      Частковий:
.
.
.
011101 = 29     AB      46: (word)   47: AB?
101111 = 47     AB?  <--- що нам з цим робити?
```

На перший погляд, для декодера це нерозв'язна задача. Ми знаємо наперед, що словом 47 повинно бути **ABA**, але як декодер дізнається про це? Відмітимо, що слово 47 складається зі слова 29 плюс символ, що йде наступним. Таким чином, слово 47 закінчується на «символ, що йде наступним». Але, оскільки це слово посилається негайно, то воно повинно починатися з «символу, що йде наступним», і тому воно закінчується тим же символом, яким і починається, в даному випадку — **A**. Цей трюк дозволяє декодеру визначити, що слово 47 це **ABA**.

В загальному випадку така ситуація з'являється, коли кодується послідовність виду $cScSc$, де c — це один символ, а S — рядок, причому слово cS вже є в словнику.

Патенти

На алгоритм LZW і його варіації був виданий ряд патентів, як в США, так і в других країнах. На LZ78 був виданий американський патент U.S. Patent 4 464 650, що належить Sperry Corporation, яка пізніше стала частиною Unisys Corporation. На LZW в США були видані два патенти: U.S. Patent 4 814 746, що належить IBM і патент Велча U.S. Patent 4 558 302 (виданий 20 червня 1983 року), що належить Sperry Corporation, пізніше перейшов до Unisys Corporation. На даний момент терміни всіх патентів збігли.

Unisys, GIF і PNG

При розробці формату GIF в CompuServe не знали про існування патенту U.S. Patent 4 558 302. В грудні 1994 року, коли в Unisys стало відомо про використання LZW в широковживаному графічному форматі, ця компанія розповсюдила інформацію про свої плани по стягненню ліцензійних відрахувань з комерційних програм, які мають можливості створення GIF-файлів. В той час формат був вже настільки широко розповсюджений, що більшість компаній-виробників ПЗ не мали іншого виходу крім як заплатити. Ця ситуація стала однією з причин розробки графічного формату PNG (неофіційна розшифровка: «PNG’s Not GIF»), який став третім по поширеності після GIF і JPEG. Наприкінці серпня 1999 року Unisys перервала дію безкоштовних ліцензій на LZW для безкоштовного та некомерційного ПЗ, а також для користувачів неліцензійних програм, закликаючи League for Programming Freedom розгорнути кампанію «спалимо всі GIF’и» й інформувати публіку про існуючі альтернативи. Багато експертів в області патентного права відмічали, що патент не розповсюджується на пристрої, які можуть лише розтискати LZW-дані, але не стискати їх; з цієї причини популярна утиліта gzip може читати Z-файли, але не записувати їх.

20 червня 2003 року закінчився термін оригінального американського патенту, тобто це означає, що Unisys не може більше стягувати по ньому ліцензійні відрахування. Аналогічні патенти в Європі, Японії та Канаді закінчились в 2004 році.

3. КОДИ З ВИЯВЛЕННЯМ І ВИПРАВЛЕННЯМ ПОМИЛОК

До найбільш поширених кодів з виявленням та виправленням помилок відносять код Хемінга, циклічні та рекурентні коди. Розглянемо їх більш детально.

Код Хемінга відноситься до систематичних кодів, в яких з п символів, які утворюють комбінацію, n_0 символів є інформаційними, а останні $k = n - n_0$ є надлишковими (контрольними), призначеними для перевірки (контрольні символи у всіх комбінаціях займають однакові позиції). Коди Хемінга дозволяють виправити всі одиничні помилки (при кодовій відстані d=3) і визначити всі подвійні помилки (при $d=4$), але не виправляти їх.

Зв'язок між кількістю інформаційних та контрольних символів в коді Хемінга знаходять на основі таких міркувань. При передачі комбінації по каналу з шумами може бути спотворений довільний з п символів коду, або комбінація може бути передана без спотворень. Таким чином може бути $n + 1$ варіантів спотворення (включаючи передачу без спотворення). Використовуючи контрольні символи, необхідно перевірити всі $n+1$ варіантів. За допомогою контрольних символів k можна описати $2k$ подій. Для цього повинна бути використана умова:

$$2^k \geq n + 1 = n_0 + k + 1 .$$

В таблиці 3.1 подана залежність між k і n_0, яка отримана з цієї невірності, де k - число контрольних символів в коді Хемінга, n_0 - інформаційних символів.

Таблиця 3.1 – Розміщення контрольних символів в комбінаціях коду Хемінга

n_0	1	2	3	4	5	6	7	8	9	10	11	12	13	14	15
k	2	3	3	3	4	4	4	4	4	4	4	5	5	5	5

В коді Хемінга контрольні символи розташовують на місцях, кратних степеню числа 2, тобто на позиціях 1, 2, 4, 8 і т. п. Інформаційні символи розташовують на місцях, що залишилися. Наприклад, для семиелементної закодованої комбінації можна записати

$$\begin{array}{ccccccc} k_1 & k_2 & a_{04} & k_3 & a_{03} & a_{02} & a_{01} \\ \boxed{a_1} & \boxed{a_2} & a_3 & \boxed{a_4} & a_5 & a_6 & a_7 \end{array} .$$

Символи коду Хемінга, які обведені прямокутниками, є *контрольними*, останні – *інформаційні*, де a_3 – старший (четвертий) розряд вихідної кодової комбінації двійкового коду, який необхідно кодувати, a_7 – молодший (перший) розряд. Після розташування на відповідних місцях кодової комбінації контрольних і інформаційних символів в коді Хемінга складають спеціальні перевірні рівняння, які використовують для визначення наявності спотворень і їх виправлення. З перевірних рівнянь і отримують контрольні символи при кодуванні вихідної кодової комбінації двійкового коду. Для визначення контрольних символів необхідно використати такий алгоритм.

1. Всі символи коду Хемінга з номерами розрядів розташовують в порядку збільшення номерів і під ними записують номери розрядів в двійковому коді

a_1 a_2 a_3 a_4 a_5 a_6 a_7

0001 0010 0011 0100 0101 0110 0111

2. Перше перевірне рівняння складають як суму за mod 2 всіх розрядів, в номерах яких в молодшому розряді 2^0 стоїть одиниця:

$$S_1 = a_1 \oplus a_2 \oplus a_5 \oplus a_7 .$$

Друге перевірне рівняння складають як суму за mod 2 всіх розрядів, в номерах яких стоїть одиниця на другому місці відповідного двійкового еквівалента (2^I):

$$S_2 = a_2 \oplus a_3 \oplus a_6 \oplus a_7 .$$

Третє перевірне рівняння складають як суму за mod 2 всіх розрядів, в номерах яких стоїть одиниця на третьому місці (2^2):

$$S_3 = a_4 \oplus a_5 \oplus a_6 \oplus a_7 .$$

Аналогічно утворюються і інші перевірні суми (при більшій кількості інформаційних і контрольних символів, відповідно).

Як видно з наведених рівнянь, в кожну перевірну суму входить тільки один невизначений контрольний символ k_i (a_1, a_2, a_4, відповідно), а всі інші інформаційні символи відомі

Всі перевірні рівняння за умовою Хемінга повинні дорівнювати 0 при підсумовуванні за mod 2. З цієї умови і знаходять контрольні символи.

Наприклад, необхідно передати інформаційну кодову комбінацію:

a1 a2 a3 a4

1 1 0 0

з числом розрядів $n_0 = 4$.

З формули $2^k /n +1 = n_0+k +1$ визначимо, що $k =3$. Запишемо перевірні суми:

$$S_1 = a_1 + a_3 + a_5 + a_7\ ;$$

$$S_2 = a_2 + a_3 + a_6 + a_7\ ;$$

$$S_3 = a_4 + a_5 + a_6 + a_7\ .$$

Підставимо у рівняння значення відомих інформаційних символів. З умови рівності нулю всіх перевірних сум визначаємо відповідно контрольні символи. З першого рівняння $a_0 = 0$, з другого $a_1 = 1$, з третього $a_4 = 1$.

Відповідно буде передана така комбінація:

$\boxed{a_1}$	$\boxed{a_2}$	a_3	$\boxed{a_4}$	a_5	a_6	a_7
0	1	1	1	1	0	0,

яка є комбінацією коду Хемінга.

Алгоритм декодування коду Хемінга

На приймальному пристрої комбінація коду Хемінга декодується - визначається наявність спотворення прийнятої кодової комбінації і, якщо один символ комбінації спотворений, він автоматично спочатку визначається, а потім виправляється. Визначення і виправлення спотвореного символу здійснюється на основі перевірних рівнянь. При правильному прийомі всі суми повинні дорівнювати нулю. В разі спотворення двійкове число, яке є результатом цих сум (синдром помилки), переводять в десяткове число, яке вказує номер спотвореного символу, який в подальшому виправляється шляхом інвертування.

Наприклад, при прийомі закодованої вище неспотвореної кодової комбінації Хемінга:

a_1	a_2	a_3	a_4	a_5	a_6	a_7
$\boxed{0}$	$\boxed{1}$	1	$\boxed{1}$	1	0	0,

значення всіх символів комбінації підставляють на відповідні місця у перевірних сумах:

$$S_1 = a_1 + a_3 + a_5 + a_7 = 0\ ;$$

$S_2 = a_2 + a_3 + a_6 + a_7 = 0$;

$S_3 = a_4 + a_5 + a_6 + a_7 = 0$.

Ознака правильно прийнятої комбінації – рівність нулю всіх сум.

Припустимо, що шостий символ кодової комбінації спотворений, тобто замість комбінації 0111100 буде прийнята:

$a_1\ a_2\ a_3\ a_4\ a_6\ a_7\ a_8$

0 1 1 1 1 1 0

Підставимо значення прийнятих символів у перевірні рівняння.

Одержимо:

$S_1 = a_1 + a_3 + a_5 + a_7 = 0$;

$S_2 = a_2 + a_3 + a_6 + a_7 = 0$;

$S_3 = a_4 + a_5 + a_6 + a_7 = 0$.

Переведемо двійкове число (синдром) $1^{2^2}1^{2^1}1^{2^0}$ в десяткове. Одержимо десяткове число 6, яке і вказує на номер спотвореного символу.

Кодер коду Хемінга

На основі наведених вище правил будується кодер коду Хемінга (рис3.2). Він автоматично визначає значення контрольних символів коду при відомих інформаційних, які складаються з безнадлишкових комбінацій звичайного двійкового коду.

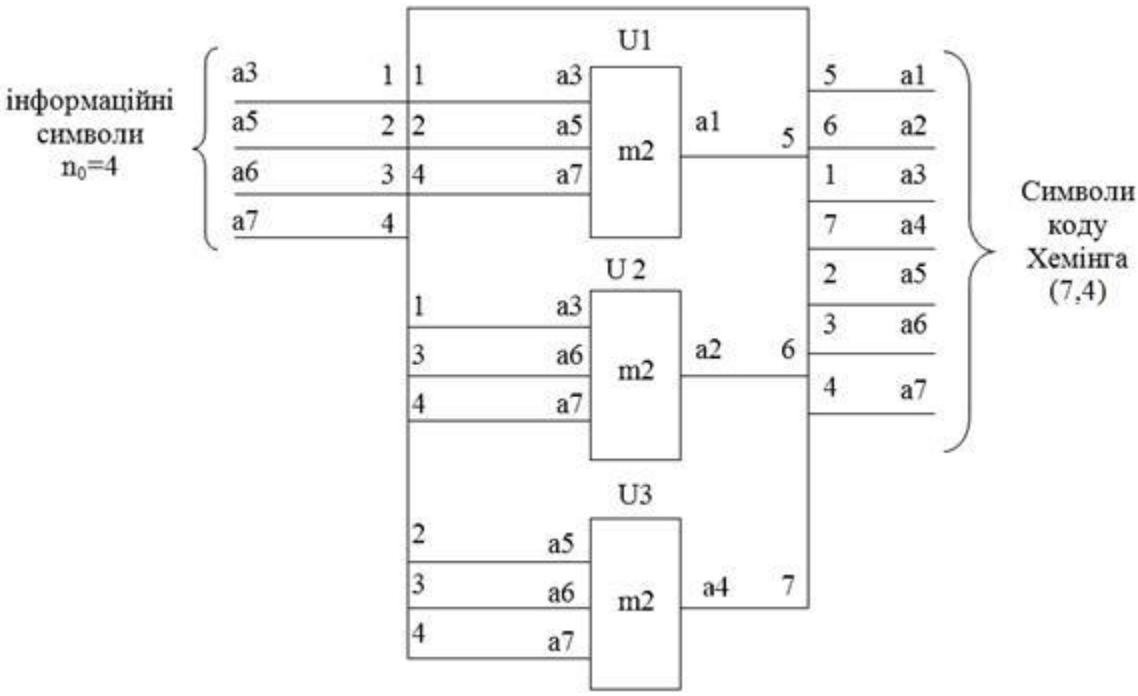

Рисунок 3.2 – Кодер коду Хемінга

На входи суматорів за mod 2 Ul, U2, U3 подаються інформаційні символи відповідно до перевірних сум. На вході суматорів одержуються контрольні символи a_1, a_2, a_4. Інформаційні і контрольні символи розташовуються в необхідному порядку і через шину виводяться для подальшого перетворення.

Декодер коду Хемінга

Декодер коду Хемінга (рис. 3.3) аналізує прийняту кодову комбінацію і в разі спотворення будь-якого одного символу (інформаційного чи контрольного) автоматично виправляє спотворений символ.

Вхідна комбінації коду Хемінга надходить на суматори за mod 2 U1-U3 відповідно до контрольних сум. На виходах суматорів утворюється результат контрольних сум S1-S3. При правильному прийомі суми SI, S2, S3 повинні дорівнювати нулю.

В разі спотворення одного символу на виходах елементів U1-U3 з'явиться двійкова кодова комбінація (синдром помилки), яка декодується декодером U4. На одному з виходів декодера з'являється рівень логічної одиниці, який відповідає номеру спотвореного символу.

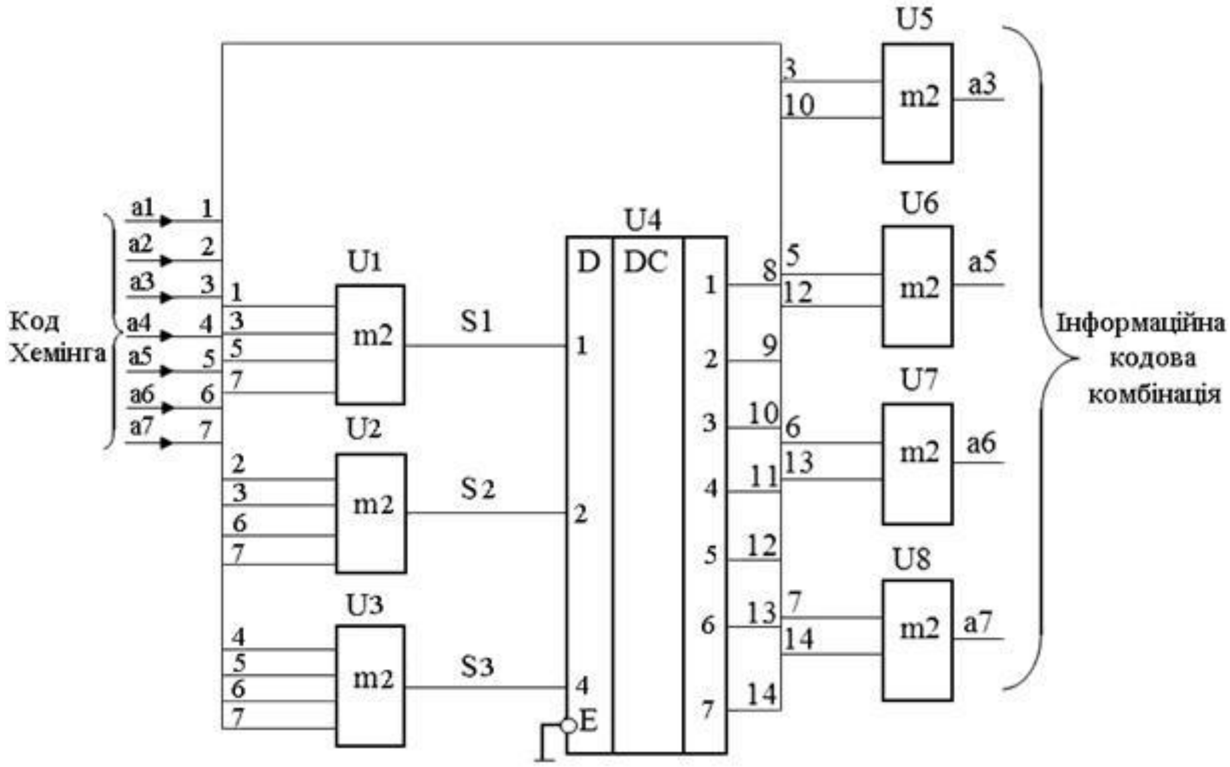

Рисунок 3.3 – Декодер коду Хемінга для семиелементної комбінації

На суматори за mod 2 U5-U8 надходять відповідні сигнали з входів a_3, a_5, a_6, a_7 (інформаційні символи) і відповідний вихід декодера.

При правильному прийомі на виходах 1-7 декодера - логічні нулі і на виходах суматорів U5-U8 з'являться інформаційні символи a3, a5 , a6 , a7 без змін.

В разі спотворення одного з символів, наприклад, третього, на третьому виході декодера з'явиться рівень логічної одиниці і на виході суматора за mod2 U5 спотворений символ автоматично інвертується, перетворюючись у правильний.

Перелік питань на іспит

1. Методологічні засади теорії інформації
2. Основні технології передачі інформації
3. Способи передачі та отримання інформації за допомогою комп’ютерних технологій
4. Інформація та її характеристики
5. Ентропія джерела інформації
6. Передача інформації без завад
7. Передача інформації із завадами
8. Перетворення Фур’є
9. Різні методи стиску інформації
10. Алгоритм Хаффмана
11. Алгоритм Шеннона-Фано
12. Арифметичний алгоритм
13. Алгоритм Лемпеля – Зіва – Велча
14. Коди з виявленням і виправленням помилок
15. Історія розвитку теорії інформації
16. Будова повідомлення
17. Концепції теорії інформації
18. Протокол Діффі-Хеллмана передачі інформації.
19. Тест Мюллера-Рабіна на простоту числа.
20. Метод Р-Полларда факторизації великого числа.
21. Методи стиснення зображень.
22. Аналого-цифрові перетворювачі сигналів.
23. Поняття про електронну систему та її характеристики
24. Класифікація електронних систем
25. Оптико-електронні системи, основні тенденції розвитку
26. Приклади електронних систем
27. Типова система передачі даних
28. Канали зв'язку. Аналогові і цифрові канали
29. Оптимальне приймання та завадостійкість
30. Виявлення та приймання інформації

Література

1. Столлингс В. Передача данных. − СПб.: Питер, 2004. − 752 с.
2. Миллер М.А., Смирнов А.И. Линии передачи . Физическая энциклопедия. Т. 2. − М.: Сов. энциклопедия, 1990. − 596 с.
3. Фред Халсалл. Передача данных, сети компьютеров и взаимосвязь открытых систем. − М.: Радио и связь, 1995. – 591 с.
4. Иртегов Д. В. Введение в сетевые технологии.— СПб.: BHV, 2004 г. – 560 с.
5. Зюко А. Г., Кловский Д.Д.и др. Теория электрической связи / Под ред. Д. Д. Кловского — Учебник для ВУЗов. — М.: Радио и связь, 1999. — С. 224. — 432 с.
6. Скляр Б. Цифровая связь. Теоретические основы и практическое применение. — Пер. с англ. — М.: Издательский дом «Вильямс», 2003. — 1104 с.
7. Советов Б. Я., О. И. Кутузов О. И. Применение микропроцессорных средств в системах передачи информации: Учеб. пособие для вузов по спец. АСУ.— М.: Высш. шк., 1990. —256 с: ил.
8. Гроднев И.И. Волоконно-оптические линии связи. Учебное пособие для ВУЗов. 2-е издание переработанное и дополненное. − М.: Радио и связь, 1990 г. − 224 с.
9. Гроднев И.И., Мурадян А.Г. и др.Волоконно-оптические системы передачи и кабели. Справочник . − М.: Радио и связь, 1993. − 264 с.: ил.