УДК 004: 004.65

В.М. ДЖУЛІЙ, Т.Є. БАБКІНА
Хмельницький національний університет

ПРИНЦИПИ ПОБУДОВИ АЛГОРИТМІВ ПОДАННЯ ІНФОРМАЦІЇ ТА ІНФОРМАЦІЙНИХ ПРЕПРОЦЕСОРІВ ДАНИХ

У статті розглянуто теоретичні основи стиснення інформації, питання класифікації та оцінки методів стиснення. Проведено аналіз методів стиснення інформації, що базуються на особливостях використовуваних в методах стиснення алгоритмів побудови моделей джерела і на типах використовуваних препроцесорів вхідної інформації.

The article reviews the theoretical basis for data compression, the question of classification and evaluation methods of compression. Analysis of data compression methods based on features used in the methods of compression algorithms and models on the types of sources used preprocessor input information.

Ключові слова: стиснення інформації, препроцесор даних, інформаційне джерело

Вступ

В даний час інформаційні комп'ютерні системи проникають в усі сфери діяльності людини. Обсяг одержаної і переробленої інформації, подвоюється кожних п'ять років. Зберігати, передавати і обробляти інформацію стає все складніше і складніше, незважаючи на швидке вдосконалення технічних засобів, призначених для вирішення зазначених задач.

Системи цифрового зв'язку (ISDN, цифровий сотовий зв'язок, цифрове телебачення тощо) також отримали загальне визнання і прогресують дуже швидко. Інформаційна інфраструктура розростається і ускладнюється. Цінність інформації що зберігається досить велика, тому застосовуються різні методи її захисту. Одним із найпоширеніших є резервне копіювання, застосування якого ще більше збільшує обсяг збереженої інформації. Все це обумовлює лавиноподібне збільшення кількості збереженої, переданої і оброблюваної інформації. Зростання кількісних і якісних характеристик сучасних технічних засобів передачі та зберігання інформації не встигає за потребами людства в таких засобах. Введення в дію нових високопродуктивних комунікаційних систем обходиться досить дорого.

Тому важливо з максимальною ефективністю використовувати наявні системи зберігання і передачі інформації. Для цього потрібно представляти накопичені дані оптимально, за рахунок їх кодування із мінімальною інформаційної надмірністю. Це дозволить зберігати більше інформації на тих же носіях, передавати більше інформації в одиницю часу по каналах зв'язку із тією ж пропускною здатністю.

Практичне використання методів стиснення інформації почалося в кінці 1970-х років, коли отримали широке поширення комп'ютерні системи з достатньою швидкодією та об'ємом оперативної пам'яті. Було запропоновано значну кількість схем стиснення інформації, встановлена важлива роль поняття моделі інформаційного джерела. Однак у теорії та практиці стиснення інформації залишилося ще чимало проблем. Так, наприклад, в якості моделі джерела до цього часу розглядаються тільки статистичні моделі, що не дають можливості узагальнити це поняття на випадок словникових та комбінованих схем стиснення інформації. Не розроблена загальна класифікація схем стиснення на основі особливостей, що лежать в їх основі, методів побудови моделей джерел. Існують резерви підвищення ефективності стиснення інформації регулярної структури, до яких належать, зокрема, файли реляційних баз даних. Покращення стиснення такої інформації може помітно скоротити потребу систем резервного копіювання інформації в об'ємах запам'ятовуючих пристроїв. Недостатньо досліджені і можливості побудови моделей джерел та алгоритмів стиснення, які ефективно адаптуються до різких змін характеру стисненої інформації.

Одним із важливих теоретичних досягнень в теорії оптимального кодування з'явилася вперше висловлена в 1981р. Ріссаненом і Ленгдоном [1,5] ідея поділу процесу оптимального кодування на дві частини:

- моделювання, яке служить для побудови моделі джерела і оцінювання імовірності появи символів на підставі побудованої моделі;
- власне кодування.

Це дало поштовх до розвитку методів моделювання джерел, що базуються на розгляді джерела як кінцевого автомата (finite state machine – FSM) з пам'яттю або без пам'яті. Також були запропоновані методи моделювання, засновані на марківських моделях станів випадкового процесу [2,3], методи контекстуального моделювання джерела [3,6] та деякі інші методи. Алгоритми, що реалізують ці методи, нетривіальні, використовують емпірично встановлені факти, але забезпечують високу якість стискування. Деякі з них заради отримання додаткового виграшу за оптимальності стиснення орієнтовані тільки на певні види інформації.

В останні роки були запропоновані декілька оригінальних методів стиснення інформації, заснованих на блочно-сортувальному перетворенні М. Барроуза і Д. Уиллера [5], і його узагальненні, запропонованому М.Шиндлером [6].

Чимало робіт в останні роки також присвячені оглядам різних методик стиснення інформації та їх порівнянні один з одним за різними критеріями, хоча останні досягнення в області методів кодування і моделювання джерел розглянуті в них досить поверхово і стисло.

Важливим напрямом досліджень в теорії оптимального кодування є оцінювання ентропії різних джерел інформації, з якими доводиться стикатися на практиці. Отримання відповідних оцінок дозволяє визначити, наскільки оптимально здійснюється кодування практичними реалізаціями тих чи інших методів оптимального кодування. Найбільший інтерес, представляє оцінювання ентропії природних мов, оскільки саме дані, що генеруються цими джерелами, є одним із основних класів інформації, до яких застосовуються методи стиснення інформації.

Першість у цих дослідженнях також належить Шеннону, який провів дослідження ентропії англійської мови [6]. Надалі багатьма дослідниками оцінювалася ентропія як широкого спектра природних мов, так і інших інформаційних джерел.

Реальні канали зв'язку в більшості випадків привносять спотворення (шуми і перешкоди) на передану інформацію, так що використовувана на виході каналу інформація не співпадає з переданою. Будемо вважати, що маємо справу з каналом без шуму, і інформація передається без спотворення. Це виправдовується тим, що задача стиснення інформації полягає у зменшенні їх надмірності, а методи підвищення надійності збереження і передачі інформації, навпаки, засновані на штучному підвищенні її надмірності.

На практиці при стисканні інформації, як правило, доводиться мати справу із вхідною інформацією у вигляді послідовності байтів, тому зручно вважати, що вихідне джерело має байтовий алфавіт, при цьому $|A| = 2^8$, де $|A|$ – потужність алфавіту джерела (кількість різних елементів алфавіту). У деяких випадках зручніше вважати, що вхідний потік має інший алфавіт, наприклад, при стисненні 16-бітових аудіоданих природно вважати, що вхідний алфавіт складається з двобайтових символів ($|A| = 2^{16}$). Вихідний алфавіт у більшості високоефективних універсальних методів є двійковим (бітовим) – $A = \{0,1\}$.

Інформаційне джерело є випадковим процесом з дискретним часом. У кожний момент часу відбувається подія, яка складається у генерації деякого повідомлення $S \in [A]$. У практичних питаннях стиснення інформації зручніше розглядати як випадковий процес, що полягає в генерації в кожен момент часу чергового символу повідомлення. У цьому випадку, як правило, кожна подія виявляється статистично залежною від попередніх. Стисненням інформації без втрат називається більш оптимальне кодування інформації – однозначно декодуєме.

Таким чином, стиснення інформації без втрат, як спеціальний вид кодування, володіє двома основними властивостями:

1) однозначна декодуємість (що означає повне збереження інформації, що міститься у вхідних повідомленнях);

2) зменшення надмірності (тобто, представлення тієї ж інформації меншою кількістю даних, в чому і полягає практична цінність стиснення інформації).

Стиснення інформації є оптимізацією подання інформації за рахунок вибору підходящого кодування.

Основною практичною задачею стиснення інформації є розробка алгоритмів, що дозволяють по вхідному повідомленню, згенерованому невідомим заздалегідь джерелом, побудувати схему кодування даного повідомлення кодом якомога меншої довжини, і закодувати повідомлення відповідно до побудованої схеми. Як вже зазначалося раніше, загальна задача стиснення інформації без втрат полягає в пошуку алгоритму, який кожному повідомленню $S \in [A]$ однозначно ставить у відповідність повідомлення $f(S)$ в двійковому алфавіті таким чином, що математичне очікування довжини отриманого повідомлення було мінімальним:

$$\sum_{S \in [A]} |f(S)|\, p(S) \to \min .$$

Незважаючи на велику кількість наявних методів стиснення інформації без втрат, у цьому напрямку залишаються наступні проблеми:

1. Є великий розрив між вартістю кодування, що досягається використовуваними універсальними методами стиснення інформації і теоретичною межею стиснення інформації – ентропія джерела (це встановлено, принаймні, для природних мов, ентропія яких оцінювалася багатьма дослідниками).

2. Наявні алгоритми, що наближаються за вартістю кодування до ентропії, орієнтовані на певні класи даних (до інших даних вони незастосовні) і, як правило, потребують збереження великих розмірів словників, що на практиці не завжди незручно.

3. Найбільш ефективні із наявних універсальних методів стиснення інформації (методи прогнозування за частковим співпаданням) дуже добре працюють на текстовій інформації, але на бінарній чи змішаній інформації їх ефективність помітно погіршується через недостатню адаптивність.

У зв'язку із вищевикладеним, ставиться задача розробки універсальних алгоритмів стиснення інформації, що мають стабільно високу стискуючу спроможність на основних класах вхідної інформації, яка піддається стисненню, і володіють підвищеною адаптивністю до мінливих статистичних характеристик джерел інформації.

Принципи побудови методів стиснення інформації

Принципи побудови методів стиснення інформації полягають у зменшенні надмірності їх кодування, тобто у наближенні вартості кодування до її межі – ентропії джерела. Розглянемо повідомлення

$S \in [A]$. Позначимо вартість кодування його *i*-го символу s_i (який виявився рівним $s_m \in A$) через $C_i(s_m)$, а кількість символів s_m у повідомленні через n_m. Середня вартість кодування $\overline{C}$ одного символу в повідомленні *S* дорівнює

$$\overline{C} = \sum_{m=1}^{M} p(\{s_i = s_m\})\,\overline{C}(s_m), \tag{1}$$

де $p(\{s_i = s_m\}) = \frac{n_m}{N}$ – оцінка ймовірності події $s_i = s_m$; $\overline{C}(s_m) = \frac{1}{n_m}\sum C_i(s_m)$ – оцінка середньої вартості.

Припустимо, що

$$C_i(s_m) = -\log_2 p(\{s_i = s_m\}) \tag{2}$$

Підставивши (2) в (1), отримуємо:

$$\overline{C} = -\sum_{m=1}^{M} p(\{s_i = s_m\}) \log_2 p(\{s_i = s_m\}) \tag{3}$$

Таким чином, при виконанні умови (2) вартість кодування дорівнює ентропії інформаційного джерела, тобто досягає теоретичного мінімуму, і, отже, таке кодування є оптимальним за вартістю.

Для успішного вирішення задачі стиснення інформації необхідно вирішити дві задачі:

1) задача оцінювання імовірності $p(\{s_i = s_m\})$;

2) задача кодування інформації відповідно до виразу (2).

Рішення задачі кодування відповідно до виразу (2) саме по собі не представляє складної задачі. Запропоновані Шенноном, Хаффманом [6] для цих цілей методи побудови алфавітних префіксних кодів з використанням двійкових кодових дерев можуть бути використані для цих цілей (їх перевагою є підвищена швидкодія), проте для отримання оптимального кодування інформації вони не підходять, оскільки кодують кожен символ цілим числом бітів, і, значить, дозволяють отримати оптимальне кодування лише у випадку, якщо всі ймовірності $p(\{s_i = s_m\})$ мають вид 2^{-n}, де *n* – натуральне число. Крім того, ці методи не дозволяють отримати оптимального представлення інформації, якщо вхідний алфавіт є двійковим: у цьому випадку кожен із символів кодується одним бітом незалежно від розподілу їх ймовірностей.

Однак арифметичне кодування [3] дає можливість кодувати символи кодом довільної (в тому числі нецілої) довжини, що і дозволяє при будь-яких значеннях ймовірностей $p(\{s_i = s_m\})$ кодувати символи оптимально. Абсолютно точне дотримання рівності (2) потенційно вимагає виконання арифметичних операцій із дійсними числами необмеженої точності (що на практиці нереалізовано), але запропонована в [5] методика дозволяє, обмежуючись цілочисельною арифметикою фіксованої точності, домогтися дуже точної апроксимації згаданої рівності. Ще однією перевагою арифметичного кодування є можливість простої реалізації адаптивного кодування.

Оскільки досліджується процес оптимального кодування в найбільш загальному випадку, слід вважати, що ніякими апріорними відомостями про властивості інформаційного джерела, що генерує вихідні дані для розглянутого алгоритму стиснення, ми не володіємо. Тобто до початку стиснення алгоритму невідомі ймовірності $p(\{s_i = s_m\})$, а також будь-які інші властивості джерела інформації. Таким чином, для оцінювання цих ймовірностей можна використовувати лише послідовність $\langle s_1, s_2, \ldots, s_i \rangle$ раніше згенерованих символів. Ця послідовність називається передісторією джерела. Оскільки передбачається, що джерело є стаціонарним дискретним випадковим процесом (або близьким до стаціонарного), шукані вірогідності можуть бути оцінені по передісторії. У деяких випадках більш вигідно будувати модель джерела, грунтуючись безпосередньо не на інформації з вхідного потоку, а на інформації, підданої деякому попередньому перетворенню (взаємно однозначному). Це виконує препроцесор даних. Його вихідний алфавіт *K* (він же вхідний алфавіт алгоритму моделювання і

Рис. 1. Узагальнена схема алгоритмів стиснення інформації

алгоритму стиснення інформації) може не співпадати з вхідним алфавітом. У такому випадку модель будується не в алфавіті *A* інформаційного джерела, а в деякому проміжному алфавіті *K*.

Узагальнена структурна стиснення, таким чином, має наступний вигляд (рис. 1)/

Зворотний по відношенню до алгоритму стиснення інформації алгоритм декодування стисненої інформації має наступну структурну схему (рис. 2).

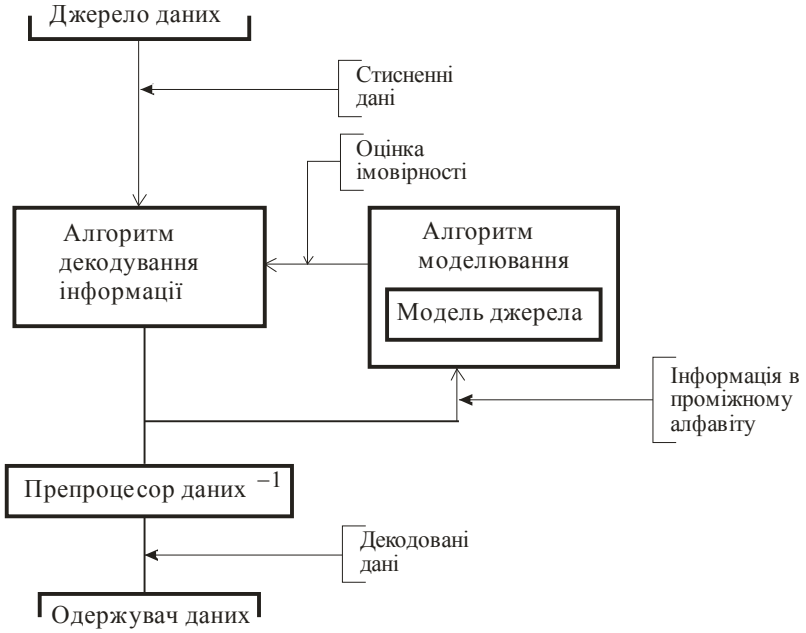

Рис. 2. Узагальнена структурна схема алгоритмів декодування стисненої інформації

Основними структурними елементами алгоритму декодування є алгоритм моделювання, алгоритм декодування інформації і препроцесор даних $^{-1}$. Алгоритм декодування здійснює перетворення, зворотне кодування (воно існує, якщо кодування взаємно – однозначне), користуючись оцінками ймовірностей символів, що видаються алгоритмом моделювання. Алгоритм моделювання абсолютно аналогічний використовуваному при стисненні інформації. Модель джерела починає будуватися з тієї ж початкової моделі, що і при стисненні інформації і синхронно модифікується символами, які надходять з виходу алгоритму декодування. Таким чином, моделі джерела при кодуванні і декодуванні ідентичні в кожен момент часу, а значить і оцінки розподілу ймовірностей символів співпадають, чим і забезпечується можливість коректного декодування. Препроцесор даних $^{-1}$ просто виконує перетворення, зворотне перетворенню препроцесора.

У статичних моделях ймовірності символів не залежать від часу, тобто джерело вважається суворо стаціонарним і в процесі стиснення ймовірність даного символу в даному контексті вважається незмінною. Оціночні значення ймовірностей або заздалегідь розраховані на інформацію певного виду (використовуються в моделі протягом всього часу її роботи), або підраховуються шляхом попереднього проходу по всьому вхідному повідомленню.

При використанні блочно – статичних моделей кодуємі повідомлення розбиваються на блоки постійної довжини N_b, всередині кожного з яких оцінки ймовірностей $p(\{s_i = s_m\})$ є постійними (розрахованими за інформацією цього блоку). Ці моделі поєднують швидкодію статичних моделей з адаптивністю і однопрохідністю адаптивних. Ступінь адаптивності можна варіювати шляхом зміни параметра N_b. Кращі реалізації таких моделей (прикладом може служити архіватор RAR) показали себе надзвичайно швидкими і досить гарними по оптимальності кодування.

У динамічних (адаптивних) моделях оцінки ймовірностей змінюються від символу до символу за рахунок того, що кожен надійшовший на її вхід символ змінює (уточнює) саму модель, робить її характеристики ближчими до поточних характеристик джерела. Такі моделі складніші і повільніші інших, але дають найбільш точні оцінки ймовірностей символів і таким чином, забезпечують оптимальне кодування.

Принципи роботи алгоритмів моделювання

Основними принципами роботи алгоритмів моделювання є побудова оцінок ймовірностей за статистичними властивостями передісторії і рівність оцінок для відповідного символу повідомлення, одержуваних алгоритмами кодування і декодування.

Будь-який алгоритм моделювання виконує дві основні функції:

1) побудова моделі джерела;

2) обчислення оцінок ймовірностей.

У статичних алгоритмах воно виконується одноразово, під час першого проходу по вхідній інформації. Отримана модель певним чином кодується і записується в початок вихідного потоку алгоритму стиснення (щоб алгоритм декодування міг отримати її до початку процесу декодування повідомлення). У блочно-статичних алгоритмах йде зчитування вхідного повідомлення блоками рівної довжини N_b кожен. Блок зберігається в оперативній пам'яті і для нього будується модель, яка кодується і записується в вихідний потік до початку кодування даних з поточного блоку. В адаптивних алгоритмах обробка здійснюється посимвольно. Перед початком стиснення є деяка початкова модель, після кодування чергового символу здійснюється оновлення моделі цим символом. Початкова модель завжди одна і та ж, тому вона відома алгоритму декодування заздалегідь, і у вихідний потік не записується. У процесі декодування також здійснюється оновлення моделі кожним декодованим символом, тому алгоритм кодування і алгоритм декодування змінюють моделі синхронно, чим і забезпечують можливість правильного декодування.

У будь-якому алгоритмі моделювання обчислення оцінок ймовірностей здійснюється на основі інформації, збереженої в моделі при її побудові, тому процес побудови моделі прямо залежить від методу, яким будуть надалі будуватися оцінки ймовірностей $p(\{s_i = s_m\})$.

Найбільш проста і природна ідея полягає в тому, щоб в якості оцінок ймовірностей $p(\{s_i = s_m\})$ взяти частоту відповідної події у передісторії.

Робота алгоритму стиснення починається з виродженої моделі, яка містить один стан. При достатньому збільшенні значення лічильника відбувається розподіл відповідного стану на два стани. Таким чином, у міру стиснення вхідного потоку інформації відбувається побудови графа станів джерела, що характеризує статистичні властивості відповідного марківського процесу.

Принципи побудови препроцесорів інформації

Розглянемо принципи побудови препроцесорів інформації. Вхідні препроцесори інформації за сферою їх застосування можна розділити на два класи:

1) універсальні препроцесори – призначені для попередньої обробки будь-якої інформації з метою, наприклад, виділення й групування повторюваної інформації або іншого перетворення статистичних характеристик вхідного потоку;

2) препроцесори, орієнтовані на вхідну інформацію – застосовуються для попередньої обробки інформації певного класу, дозволяють підвищити регулярність статистичних характеристик вхідного потоку за рахунок використання особливостей, характерних для даних цього класу.

До першого класу насамперед належать словникові препроцесори. Вони, у свою чергу, можуть бути розділені на два підкласи:

- з лінійним словниковим буфером;
- з гніздовим словником.

Препроцесори з лінійним словниковим буфером містять словник у вигляді кільцевого буфера, що містить N попередніх символів повідомлення, і кодують слова вхідного потоку (словами називаються послідовності ідучих підряд символів з вхідного потоку), як посилання у вигляді пар (зсув, довжина) на аналогічні слова, що вже містяться в буфері. Процес розбиття вхідного потоку на слова називається словниковим розбором. Такі препроцесори дозволяють будувати дуже ефективні схеми стиснення, що відрізняються високою щільністю стиснення і дуже високою швидкістю декодування.

Сімейство препроцесорів з гніздовим словником містять заповнюючий у міру стиснення вхідного повідомлення словник, що складається з нумерованих входів (гнізд), кожному з яких алгоритм словникового розбору ставить у відповідність деяке слово, яке зустрічалося раніше у вхідному потоці. В подальшому слово, що зустрілося у вхідному потоці, зі словника кодується просто відповідним йому номером словникового гнізда. Розроблено різні варіанти таких препроцесорів, що відрізняються в основному правилами словникового розбору. Перевагою таких схем є висока швидкість стиснення і декодування і порівняно невеликі вимоги до оперативної пам'яті при прийнятній якості стиснення інформації.

Крім словникових, останнім часом широко використовуються блочно-сортуючі універсальні препроцесори. Дані препроцесори здійснюють спеціального виду оборотну перестановку символів вхідного потоку всередині вікна (блоку) фіксованого розміру.

При обробці відсортованої інформації та послідовностей вимірювань використовується дельта-кодування, яке полягає в заміні послідовності елементів інформації послідовністю різниць сусідніх елементів (т.зв. дельт). Проблемою є автоматичне визначення розміру елемента вхідної інформації. У разі послідовностей вимірювань, як правило, має місце періодичність інформації, тому детектування періодів зростання і зменшення значень елементів інформації дозволяє додатково поліпшити стиснення.

Особливо велике різноманіття алгоритмів і широкі можливості препроцесорів, орієнтованих на інформацію. Практично для кожного випадку вузькоспеціалізованої вхідної інформації може бути розроблений спеціалізований препроцесор, що дозволить далі значно ефективніше застосувати один із універсальних методів стиснення інформації.

Класифікація препроцесорів вхідної інформації наведена на рис. 3. Дана класифікація базується на відмінностях у характері оброблюваної різними препроцесорами інформації, використовуваної внутрішньої

організацією інформації та особливості алгоритму препроцесора.

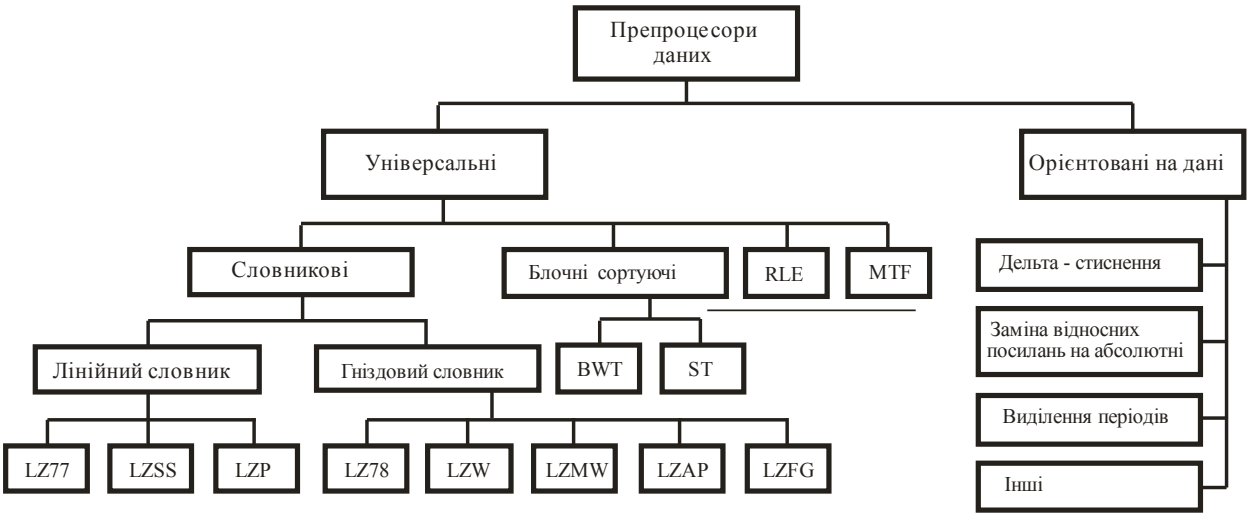

Рис. 3. Класифікація препроцесорів вхідної інформації

Комбінуючи наявні методи моделювання, кодування і побудови препроцесорів відповідно до загальної структурної схеми методів стиснення інформації (рис. 1), можна отримати широкий спектр схем стиснення інформації. Однак при побудові класифікації методів стиснення з метою уникнення її невиправданого ускладнення слід врахувати наступні моменти: далеко не будь – яке поєднання алгоритмів моделювання, алгоритмів стиснення і препроцесора є перспективним з точки зору побудови високоефективних практичних методів стиснення. Наприклад, для словниково-гніздових препроцесорів всі наявні моделі не дають істотного виграшу, а застосування будь-яких словникових препроцесорів помітно погіршує показники ефективності алгоритмів побудови моделей високих порядків; методи Хаффмана і Шеннона-Фано дають алфавітно-префіксні коди з близькою або рівною вартістю кодування, причому перший код ніколи не гірше другого (іноді – краще). Задача теоретичної оцінки потужності сучасних методів стиснення інформації надзвичайно складна і не отримала ще якого-небудь задовільного рішення.

Висновки

Розглянуто теоретичні основи стиснення інформації, питання класифікації та оцінки методів стиснення. Проведено аналіз методів стиснення інформації, що базуються на особливостях використовуваних в методах стиснення алгоритмів побудови моделей джерела і на типах використовуваних препроцесорів вхідної інформації.

Таким чином, визначено найбільш придатні базові схеми для розробки методів стиснення інформації, які мають підвищену здатність стиснення, а також напрямки, в яких слід удосконалювати кожну з них.

Література

1. Бриллюэн Л. Наука и теория информации / Бриллюэн Л.; [пер. с фр. Е.В. Гайдукова и Н.Н. Родман] – М.: Физматгиз, 1960. – 749 с.

2. Ван Тассел Д. Стиль, разработка, эффективность, отладка и испытание программ / Ван Тассел Д.; [пер. с англ. Е.К. Масловского] – М.: Мир, 1985. – 368 с.

3. Вольфовиц Дж. Теоремы кодирования теории информации / Вольфовиц Дж.; [пер. с англ. Л.Е Филипповой] – М.: Мир, 1967. – 12 с.

4. Кнут Д. Искусство программирования для ЭВМ / Кнут Д.; [пер. с англ. Н.И. Вьюковой и др. под ред. Ю.М. Баяковского и В.С. Штаркмана] – М.: Мир, 1977. – Т.2: Получисленные алгоритмы. – 725 с.

5. Файнстейн А. Основы теории информации / Файнстейн А.; [пер. с англ. И.Н. Коваленко] – М.: Иностранная литература, 1960. – 186 с.

6. Шеннон К.Э. Математическая теория связи / К.Э. Шеннон // Работы по теории информации и кибернетике. – М.: Иностранная литература, 1963. – С. 243-332.

Надійшла до редакції
11.10.2010 р.