УДК 681.3

В.А. Алексеєв, С.А.Ільїн, В.С. Терещенко, Д.А. Ференц

РЕІНЖЕНІРІНГ БАЗ ДАНИХ ІНФОРМАЦІЙНИХ СИСТЕМ З РОЗГАЛУЖЕНОЮ МЕРЕЖЕЮ ПУНКТІВ ЗБОРУ ПЕРВИННОЇ ІНФОРМАЦІЇ

Наводяться шляхи реінженірінгу баз даних автоматизованих інформаційних систем з метою упорядкування алфавітної однорідності ідентифікаційних реквізитів, введених в систему на різних мовах, та їхнього розміщення по відведених для них полях БД залежно від алфавіту представлення реквізитів для підвищення ефективності вирішення функціональних задач з пошуку інформації в системі по цих реквізитах при використанні методів чіткого пошуку.

Вступ

В автоматизованих інформаційних системах (АІС) з широко розгалуженою мережею пунктів збору первинної інформації, що пов'язана з ідентифікаційними реквізитами (прізвище, ім'я та по батькові), іноді виникають проблеми орфографічного характеру (порушуються правила, що регулюють способи передачі мови на письмі [1]), які можуть призвести до значного зниження ефективності роботи таких АІС, а то й до повного їх колапсу, коли, скажімо, система на запит у процесі інформаційного пошуку не може знайти у базі даних необхідні ідентифікаційні реквізити (ІР), хоча достеменно відомо, що вони там є. Передусім це стосується тих систем, які вже давно введені в експлуатацію. Їхні ресурси (інформаційні і програмні) започатковані давно і з часом поступово нарощувались, мережа пунктів збору збільшувалась, кількість програмних та апаратних компонентів теж. Ці компоненти могли створюватись в різні часи і різними колективами розробників. Дії цих колективів були недостатньо узгоджені між собою. В процесі розробки та експлуатації таких систем нормативи та вимоги до них могли змінюватись, але корективи не завжди вносились у вже працюючі компоненти. А ще могли виникати нагальні потреби оперативної зміни якихось компонентів, доробки, а то й розробки їх заново, що не завжди узгоджувалось з уже працюючими.

У пунктах збору спочатку могла вноситись інформація однією мовою, згодом – іншою, причому в одних задачах це могло бути враховано, а в інших ні. На одних пунктах збору первинної інформації ІР могли вноситись в одному регістрі, а на інших – в іншому. Архітектура баз даних (БД) могла змінюватись, а інформація БД неодноразово модифіковуватись. Та й автоматизовані робочі місця (АРМ) на пунктах збору, ще за часів введення системи в експлуатацію, працювали в MS DOS, де кодові таблиці не співпадають з кодовими таблицями MS Windows (наприклад, кодова таблиця No 866 та No 1251 Windows в ASCII), наявний драйвер клавіатури не завжди був розрахований на українську мову, що змушувало операторів застосовувати літери інших алфавітів. До того ж треба додати людський фактор як під час занесення громадянами даних про ІР до певних карток та анкет, так і під час введення їх операторами багаточисельних пунктів збору в систему.

Але нашою метою є не пошук причин виникнення такого становища, а пошук шляхів його виправлення. І саме цьому присвячена дана стаття. А причини тут наведені лише для більшого загострення уваги на цій проблемі та аналізу її наслідків.

1. Визначення проблеми

Дослідивши детально описану вище ситуацію, що склалася в давно вже працюючій інформаційній системі, можна зробити висновок про те, що вона може привнести в БД системи наступні негативні явища, так чи інакше пов'язані з орфографічними проблемами записів ідентифікаційних реквізитів ІР, що не дозволяють ідентифікувати ці реквізити в БД у разі використання методів чіткого пошуку:

- наявність в БД крім літер, використовуваних в системі алфавітів, ще й "заборонених" символів (цифр, розділових знаків, спеціальних символів, прогаликів перед реквізитом або кількох разом в інших місцях тощо), що не передбачені для використання в написанні ІР;
- наявність в БД однакових ІР, але введених у різних регістрах (верхній – великі літери, регістр – малі);
- наявність в ІР літер, що належать різним алфавітам або однакові за написом ІР можуть бути подані літерами різних алфавітів (наприклад, ІВАНОВ може бути написано літерами як українського, так і англійського алфавітів);
- місцезнаходження ІР одного алфавіту в полі БД, призначеного для ІР іншого алфавіту.

Це може відбуватися в системах, які оперують ІР, записаними кількома мовами. Можна сказати, в рамках системи, так би мовити, "взаємодіють" кілька алфавітів: два, три або навіть і більше. Назвемо такі системи 2-алфавітними, 3-алфавітними,...,*N*-алфавітними. Під алфавітом (абеткою) будемо розуміти сукупність літер, апостроф та прогалик, упорядкованих певним чином, на відміну від інших визначень, наприклад: сукупність літер, складових знаків та інших графем даної системи письма, розміщених у певному порядку [2], впорядкована певним образом сукупність взаємно відмінних знаків (літер, цифр, спеціальних і службових знаків) та прогалику [3], впорядкований набір усіх літер цифр та знаків конкретної мови [4], не кажучи вже про алфавіти, що використовуються в мовах програмування.

Подолати ці негативні явища можна тільки тоді, коли будуть вирішені відповідні задачі. Більше того, вирішення цих задач необхідно періодично повторювати під час, скажімо, проведення регламентних робіт на БД, тому що за наявності багатьох джерел введення інформації не можна гарантувати у подальшому появи цих явищ знову.

Першу задачу – вивільнення БД від "заборонених" при написанні ІР символів вирішити легко під час профілактичних робіт на БД інформаційної системи, вилучивши їх програмними засобами або замінивши на прогалики залежно від попередньої домовленості.

Другу задачу – зняття ускладнень з інформаційно-пошукових процесів через наявність в системі ІР в різних регістрах теж вирішити неважко. Для цього достатньо невеличкого додатку до програмних засобів пошуку, в якому ідентифікація літер проводиться незалежно від регістру, в якому вони введені та зберігаються. А ще краще (і це буде видно трохи згодом) перевести всі реквізити в БД у верхній регістр і для унеможливлення у подальшому вводу реквізитів у нижньому регістрі модифікувати програмні засоби вводу даних первинної інформації для автоматичного переводу у верхній регістр при внесенні реквізитів у систему.

Третя задача дещо складніша. Вона полягає у приведенні записів ІР у алфавітну норму, тобто зробити їх алфавітно-однорідними (всі літери запису ІР у полі БД мають бути одного алфавіту), а для цього при локалізації неоднорідності потрібно вміти програмно знаходити літеру необхідного для даного ІР алфавіту, що збігається за написанням з помилковою літерою (іншого алфавіту), щоб поміняти їхні коди. Саме такі помилки превалюють у БД. Якщо ж помилкова літера за написанням не збігається з жодною літерою необхідного алфавіту, тоді програмне виправлення неможливе. Тут потрібне втручання оператора для визначення початкового запису ІР у вхідних документах, що негативно відбивається на часі реінженірінгу БД.

Четверта задача полягає в розміщенні вже алфавітно-однорідних ІР у призначених для них залежно від алфавіту полях БД, а для цього треба вміти програмно визначати приналежність реквізиту до того чи іншого алфавіту.

Вирішення цих задач і є метою реінженірінга БД інформаційної системи, в якій для записів ІР використовуються декілька алфавітів для зняття описаної проблеми. Зрозуміло, тут перш за все йдеться про 3-алфавітні системи, як про найбільш актуальні на даний момент у нашій країні з трьома національними мовами: україн-

ська, російська та англійська. Саме вони є найбільш вживаними для заповнення деяких громадянських документів облікового характеру. Часто ці мови розділяють за письмом: кирилицею та латиницею [2], хоча, строго кажучи, це не зовсім так. Англійський алфавіт (26 літер) не співпадає на 100% з латинським письмом (25 літер), а український (33 літери) та російський (33, але дещо інші літери) алфавіти – з кириличним (43 літери) письмом. Але, не дивлячись на це, не будемо відходити від такого умовного розподілу. Тим більше що він прийнятий і в додатках середовища MS Windows, а ми розглядатимемо різницю та збіжність написання літер названих вище алфавітів не самих мов, як філологічних та лексичних засобів людського спілкування з їхніми фонетичними, морфологічними, синтаксичними особливостями [5], а їхніх представлень, саме представлень, в цих додатках та програмних засобах ведення БД.

Серед літер кожного з цих алфавітів є такі, що за написанням (графічним зображенням у кожному з прийнятих у додатках MS Windows шрифтів):

- збігаються між собою у всіх трьох алфавітах;
- збігаються в кожній парі алфавітів окремо (в кожній парі – це для загального випадку);
- притаманні саме цьому, тільки одному алфавіту (з числа тих, що розглядаються).

Перелік цих літер, як тих, що збігаються, так і тих, що не збігаються по написанню для вибраних українського, російського та англійського алфавітів у верхньому регістрі, наведено у табл. 1. Зрозуміло, що для нижнього регістру таблиця буде іншою, а тому для зменшення складності задачі і трудомісткості її вирішення і пропонувалося вище вводити та зберігати IP в одному (верхньому) регістрі.

Таблиця наведена для 3-алфавітної системи, для конкретного випадку – тільки для вказаних алфавітів. Розглянемо більш загальний випадок – 3-алфавітну систему з трьома будь-якими національними алфавітами, застосувавши при цьому формальний опис “взаємодіючих”, в рамках інформаційної системи та сформульованої вище проблеми, алфавітів.

2. Формальний опис взаємодіючих алфавітів

Нехай M_1, M_2, M_3 – множини літер національних алфавітів відповідно A_1, A_2, A_3, а $b_i(A_1)$, $b_\alpha(A_2)$, $b_\beta(A_3)$ – елементи цих множин (літери вказаних алфавітів):

$$b_i(A_1) \in M_1 \quad \text{для} \quad i \in \{1, 2, ..., L_1\};$$

$$b_\alpha(A_2) \in M_2 \quad \text{для} \quad \alpha \in \{1, 2, ..., L_2\};$$

$$b_\beta(A_3) \in M_3 \quad \text{для} \quad \beta \in \{1, 2, ..., L_3\};$$

де L_1, L_2, L_3 – кількість елементів в множинах M_1, M_2, M_3 відповідно.

Теоретико-множинне об’єднання цих множин (так звана “взаємодія” алфавітів A_1, A_2, A_3 в рамках проблеми й

Таблиця 1. **Перелік літер, що збігаються і не збігаються по написанню для українського, російського та англійського алфавітів у верхньому регістрі**

Літери, що по написанню не збігаються з літерами інших алфавітів, а є тільки в наступних алфавітах			Літери, що по написанню збігаються тільки в наступних парах алфавітів			Літери, що збігаються в усіх трьох алфавітах
українському	російському	англійському	українському, російському	українському, англійському	російському, англійському	
Ґ, Є, Ї	Ё, Ъ, Ы, Э	D, F, G, J, L,	Б, Г, Д, Ж, З,	І	-	A, B, E, K, M,
		N, Q, R, S, U,	И, Й, Л, П, У,			H, O, P, C, T,
		V, W, Y, Z	Ф, Ц, Ч, Ш, Щ,			X
			Ь, Ю, Я			

інформаційної системи) породжує літерний (символьний) простір системи – нову, так би мовити, просторову множину M, елементами якої є всі літери всіх алфавітів, задіяних у системі. Можна сказати, що ці елементи складають 3-вимірний простір літер, вимірами якого є кожний із "взаємодіючих" алфавітів як поняття, а не сукупність літер. Ця просторова множина є сумою цілого сімейства підмножин [6, с. 18], що виникають через взаємне комбіноване перерізання заданих множин: кожна з кожною попарно, усі разом, ще й різниці просторової множини з перерізаннями заданих множин. Для формального запису цього утворення введемо символьне позначення таких підмножин – M_w^v. Воно має два індекси: верхній *v* вказує на кількість множин, підмножиною яких є позначена підмножина, і нижній *w* – на конкретні множини, підмножиною саме яких є позначена підмножина. Наведемо це утворення із залученням описаного символьного позначення підмножин:

$$M \Leftrightarrow M_1 \bigcup M_2 \bigcup M_3 \Leftrightarrow$$
$$\Leftrightarrow M_1^1 \oplus M_2^1 \oplus M_3^1 \oplus M_{12}^2 \oplus M_{13}^2 \oplus M_{23}^2 \oplus M_{123}^3,$$

де M_1^1 – підмножина множини M_1 ($M_1^1 \subset M_1$) літер тільки в алфавіті A_1;

M_2^1 – підмножина множини M_2 ($M_2^1 \subset M_2$) літер тільки в алфавіті A_2;

M_3^1 – підмножина множини M_3 ($M_3^1 \subset M_3$) літер тільки в алфавіті A_3;

M_{12}^2 – підмножина двох множин: M_1 та M_2 ($M_{12}^2 \subset M_1$, $M_{12}^2 \subset M_2$) літер алфавітів A_1, A_2, які за написом збігаються;

M_{13}^2 – підмножина двох множин: M_1 та M_3 ($M_{13}^2 \subset M_1$, $M_{13}^2 \subset M_3$) літер алфавітів A_1, A_3, які за написом збігаються;

M_{23}^2 – підмножина двох множин: M_2 та M_3 ($M_{23}^2 \subset M_2$, $M_{23}^2 \subset M_3$) літер алфавітів A_2, A_3, які за написом збігаються;

M_{123}^3 – підмножина трьох множин: M_1, M_2, M_3 ($M_{123}^3 \subset M_1$, $M_{123}^3 \subset M_2$, $M_{123}^3 \subset M_3$) літер алфавітів A_1, A_2, A_3, які за написом збігаються.

Розподіл вказаних підмножин по графах табл. 1 для вже дослідженого нами випадку (якщо прийняти український алфавіт за A_1, російський за A_2, а англійський за A_3) буде наступним: M_1^1 – графа 1, M_2^1 – графа 2, M_3^1 – графа 3, M_{12}^2 – графа 4, M_{13}^2 – графа 5, M_{23}^2 – графа 6, M_{123}^3 – графа 7.

Схематичне зображення об'єднання множин M_1, M_2, M_3, їхнього комбінованого взаємного перерізання та породжених цим дійством підмножин M_1^1, M_2^1, M_3^1, M_{12}^2, M_{13}^2, M_{23}^2, M_{123}^3 показане на рис.1.

Кожній літері відповідає певний код з кодової сторінки, а множині – область визначення кодів літер відповідного алфавіту A_1, A_2 або A_3 на кодовій сторінці (наприклад, кодова сторінка No 1251 Windows для Power Builder 8.03 стандартної системи кодування ASCII). Відомо, що на цій кодовій сторінці можна виділити область визначення кодів для літер алфавітів A_1, A_2 та A_3. Деякі області визначення кодів літер можуть перерізатися або навіть бути єдиними для кількох алфавітів (наприклад, для українського та російського). Деякі області визначення кодів літер можуть бути різними, хоча відповідні їм літери збігаються за написанням (наприклад, український та англійський). Це вносить додаткові складнощі в задачу розпізнавання належності літери до того чи іншого алфавіту.

3. Перевірка алфавітної однорідності ідентифікаційного реквізиту

Перевірка алфавітної однорідності IP зводиться до аналізу (визначення) належності літер цього IP до досліджених

вище семи підмножин: M_1^1, M_2^1, M_3^1, M_{12}^2, M_{13}^2, M_{23}^2, M_{123}^3 і залежно від цього робиться висновок про подальші дії:

– поставити позначку в БД про алфавітну однорідність цього запису ІР, щоб при наступній перевірці не перевіряти його знов;

– перенести запис ІР в інше призначене для нього поле, якщо він однорідний, але знаходиться не на "своєму" місці в БД та поставити позначку;

– поміняти код літери, якщо вона хоча і збігається по написанню, але належить іншому алфавіту, та поставити позначку;

– провести організаційні дії, якщо в записі ІР застосовано літери іншого алфавіту, навіть з іншим графічним зображенням, виправити які можна, тільки перевіривши первинні документи, де вперше з'явився цей ІР, та відкоригувавши його вручну.

Таким чином, варіантів перевірки однорідності може бути тільки сім по кількості підмножин. В кожному з них може бути три випадки стосовно підмножини, що досліджується саме в цьому варіанті:

- 1: всі літери ІР належать цій підмножині;
- 2: деякі літери ІР належать цій підмножині;
- 3: жодна з літер ІР не належить цій підмножині.

Третій випадок по кожному із семи варіантів розглядати нема сенсу, тому що він характерний для кожного варіанту і дії щодо нього необхідні однакові: потрібно перевірити інші варіанти, тому що цій підмножині ІР не належить.

Розглянемо варіанти визначення належності літер ІР до вказаних підмножин. Наприклад, перевіримо по всіх можливих семи варіантах ІР, що знаходиться у полі, призначеному для ІР, написаних, скажімо, в алфавіті A_1:

➢ *варіант 1* – аналіз літер ІР щодо належності їх підмножині M_1^1:

- випадок 1: всі літери ІР належать підмножині M_1^1 – ІР є алфавітно однорідним і належить алфавіту A_1;
- випадок 2: деякі літери ІР належать підмножині M_1^1 – ІР є алфавітно неоднорідним і необхідно дослідити інші літери, що не належать алфавіту A_1, відповідно до шляхів, викладених у подальших варіантах, визначивши перед тим, до якої підмножини належать ці літери;

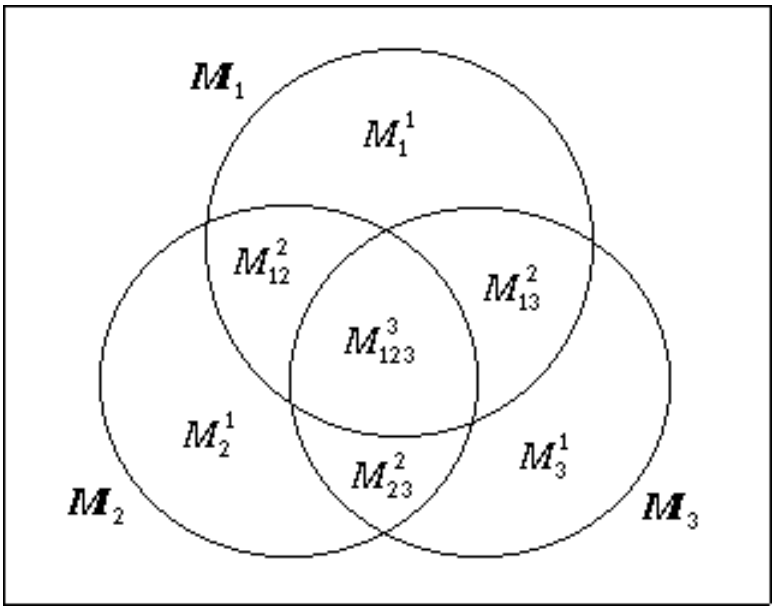

Рис. 1. Схематичне зображення об'єднання множин літер алфавітів в 3-алфавітній системі

- *варіант 2* – аналіз літер ІР щодо належності їх підмножині M_2^1:
  - випадок 1: всі літери ІР належать підмножині M_2^1 – ІР знаходиться не у своєму полі БД і його треба перенести до поля ІР з літерами алфавіту A_2;
  - випадок 2: деякі літери належать підмножині M_2^1 – в ІР є граматичні помилки (застосовано літери іншого алфавіту, навіть з іншим графічним зображенням), виправити які можна тільки через організаційні дії (перевірку первинних документів, де вперше з'явився цей ІР, та коригування його вручну);
- *варіант 3* – аналіз літер ІР щодо належності їх підмножині M_3^1:
  - випадок 1: всі літери ІР належать підмножині M_3^1 – ІР знаходиться не у своєму полі БД і його треба перенести до поля ІР з літерами алфавіту A_3;
  - випадок 2: деякі літери належать підмножині M_3^1 – висновки ті ж, що й у попередньому варіанті;
- *варіант 4* – аналіз літер ІР щодо належності їх підмножині M_{12}^2:
  - випадок 1: всі літери належать підмножині M_{12}^2 – тут потрібна додаткова перевірка кодів цих літер: якщо вони відносяться до області визначення кодів алфавіту A_1, то ІР є алфавітно-однорідним; якщо до алфавіту A_2, то треба змінити код літери на відповідний код цієї літери з області визначення кодів літер алфавіту A_1;
  - випадок 2: деякі літери належать підмножині M_{12}^2 – шлях той же, що й у попередньому випадку цього варіанта;
- *варіант 5* – аналіз літер ІР щодо належності їх підмножині M_{13}^2:
  - випадок 1: всі літери належать підмножині M_{13}^2 – висновки ті ж, що й у попередньому варіанті, але відносно алфавіту A_3;
  - випадок 2: деякі літери належать підмножині M_{13}^2 – шлях той же, що й у попередньому пункті цього варіанта;
- *варіант 6* – аналіз літер ІР щодо належності їх підмножині M_{23}^2:
  - випадок 1: всі літери ІР належать підмножині M_{23}^2 – ІР знаходиться не у своєму полі БД і його треба перенести або до поля A_2, або до поля A_3, замінивши коди літер, що знаходяться в області визначення кодів літер не "свого" алфавіту, на відповідні коди цих літер "свого" алфавіту;
  - випадок 2: деякі літери ІР належать підмножині M_{23}^2 – висновки ті ж, що й у випадку 2 варіанта 2 та 3;
- *варіант 7* – аналіз літер ІР щодо належності їх підмножині M_{123}^3:
  - випадок 1: всі літери належать підмножині M_{123}^3 – тут потрібна додаткова перевірка кодів цих літер: якщо вони відносяться до області визначення кодів алфавіту A_1, то ІР є алфавітно-однорідним, якщо до алфавіту A_2 або A_3, то треба змінити код літери на відповідний код цієї літери з області визначення кодів літер алфавіту A_1;
  - випадок 2: деякі літери належать підмножині M_{123}^3 – шлях той же, що і в попередньому випадку цього варіанта.

Перевірка завершена: досліджені всі можливі варіанти, тобто всі підмножини для вибраної кількості алфавітів. Для інших полів ІР викладки будуть ідентичними.

Визначення належності алфавітно-однорідного ІР до певного алфавіту A_1,

A_2, A_3 базується на визначенні належності його літер як елементів до множин M_1, M_2, M_3.

4. Розширення задачі на довільну кількість алфавітів

Читачі, мабуть, помітили, що склад підмножин M_1^1, M_2^1, M_3^1, M_{12}^2, M_{13}^2, M_{23}^2, M_{123}^3 має деякі особливості. Їх можна розділити на три групи залежно від кількості множин, представлених своїми елементами в цих підмножинах. До першої групи віднесемо M_1^1, M_2^1, M_3^1, тому що кожна з них є підмножиною тільки однієї множини. До другої групи – підмножини M_{12}^2, M_{13}^2, M_{23}^2: кожна з них є підмножиною двох множин. До третьої групи – підмножину M_{123}^3: вона є підмножиною усіх трьох множин. Таким чином, 3-алфавітна "взаємодія" породжує три групи підмножин.

Неважко визначити, що в інформаційній *N*-алфавітній системі "взаємодія" *N* алфавітів приведе до створення сімейства підмножин множини літерного (символьного) простору M системи, яке можна представити у вигляді *N* груп підмножин. В першій групі – підмножини, кожна з яких є підмножиною тільки однієї множини (містить окремі літери тільки одного алфавіту, *v*=1), у другій групі – підмножини, кожна з яких є підмножиною двох множин (містить окремі літери двох алфавітів, *v*=2) і так далі, в *v*-й групі – підмножини, кожна з яких є підмножиною *m* множин (містить окремі літери *m* алфавітів, *v*=*m*), в *N*-й групі завжди буде одна підмножина (містить окремі літери всіх *N* алфавітів, *v*=*N*). Кількість підмножин K_N^v у *v*-й групі (*v*=*m*) визначається математичним сполученням із *N* елементів по *m* ($m \in \{1, 2, ..., N\}$), а загальна кількість підмножин K_N дорівнює

$$K_N = \sum_{v=1}^{N} K_N^v = \sum_{m=1}^{N} C_N^m ,$$

де C_N^m – комбінаторна функція [7] математичного сполучення із *N* елементів по *m*;

N – кількість множин (кількість алфавітів в системі);

v – номер групи підмножин ($v \in \{1, 2, ..., N\}$);

m – кількість множин, підмножиною яких є дана підмножина (в даному разі кількість множин та номер групи підмножин збігаються – *m* = *v*);

K_N^v – кількість підмножин в *v*-й групі підмножин (кожна з підмножин цієї групи містить окремі літери *m* алфавітів з *N* алфавітів, які є в системі, для *v* = *m*).

При збільшенні *N* кількість підмножин K_N^v по групах дуже швидко зростає. Це видно навіть поверховим оглядом при порівнянні рис.1, 2 та 3.

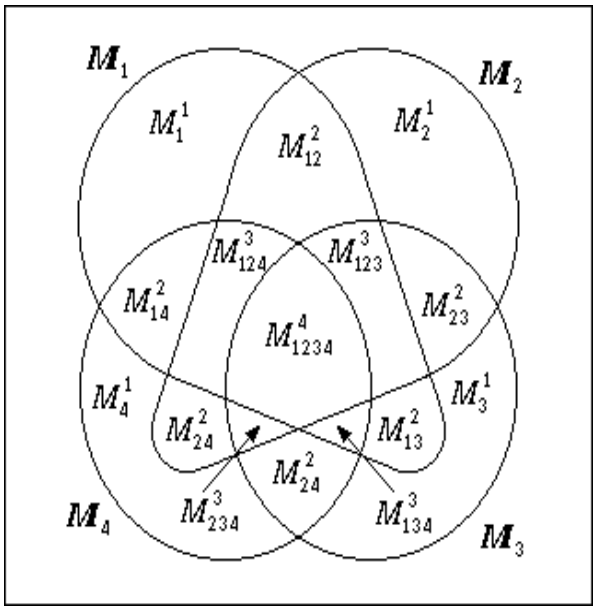

Рис. 2. Схематичне зображення об'єднання множин літер алфавітів в 4-алфавітній системі

На рис. 2 показане схематичне зображення об'єднання 4-х множин M_1, M_2, M_3, M_4, їхнього комбінованого взаємного перерізання та породжених цим дійством підмножин по групах: K_4^1=4 (M_1^1, M_2^1, M_3^1, M_4^1), K_4^2=6 (M_{12}^2, M_{13}^2, M_{14}^2, M_{23}^2, M_{24}^2, M_{34}^2), K_4^3=4 (M_{123}^3, M_{124}^3, M_{134}^3, M_{234}^3), K_4^4=1 (M_{1234}^4).

На рис. 3 показане схематичне зображення об'єднання 5-ти множин M_1, M_2, M_3, M_4, M_5, їхнього комбінованого взаємного перерізання та породжених цим дійством підмножин по групах: K_5^1=5 (M_1^1, M_2^1, M_3^1, M_4^1, M_5^1), K_5^2=10 ($M_{12}^2, M_{13}^2, M_{14}^2, M_{15}^2, M_{23}^2, M_{24}^2, M_{25}^2, M_{34}^2, M_{35}^2, M_{45}^2$), K_5^3=10 ($M_{123}^3, M_{124}^3, M_{125}^3, M_{134}^3, M_{135}^3, M_{145}^3, M_{234}^3, M_{235}^3, M_{245}^3, M_{345}^3$), K_5^4=5 (M_{1234}^4, M_{1235}^4, M_{1245}^4, M_{1345}^4, M_{2345}^4), K_5^5=1 (M_{12345}^5).

На рис. 3 деякі підмножини зустрічаються двічі, але це не значить, що таких підмножин і у загальній сукупності об'єднання кілька. Це тільки прояв особливостей графічного зображення схеми.

В табл. 2 наведена кількість підмножин по групах як результат комбінованого взаємного перерізання початкових множин, для N-алфавітних, де N={1, 2, …, 9}, систем.

Ця таблиця являє собою фрагмент трикутника Паскаля з біноміальними коефіцієнтами, а взаємодію N алфавітів, вірніше, їхніх множин, в системі можна розглядати як розклад біному Ньютона, степінь якого дорівнює N, у повній відповідності до основ класичної математики і, зокрема, теорії множин та комбінаторики.

Стосовно викладок щодо шляхів вирішення поставленої задачі для N-алфавітних систем, то вони будуть аналогічними, але з відповідними корективами кількості підмножин та груп цих підмножин.

5. Практична реалізація запропонованих шляхів вирішення задачі

Наведені в статті викладки знайшли практичне втілення у одному з про-

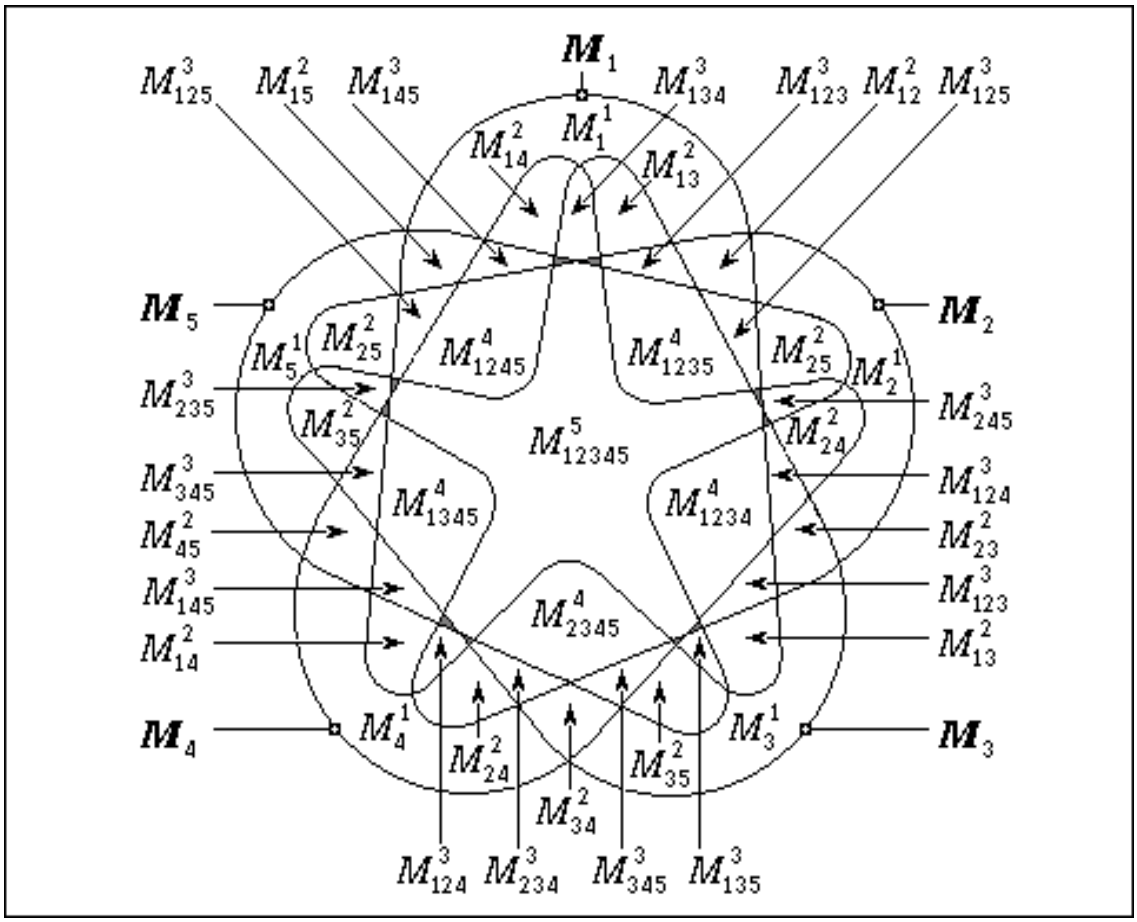

Рис. 3. Схематичне зображення об'єднання множин літер алфавітів в 5-алфавітній системі

Таблиця 2. **Кількість підмножин (по групах і загалом) для *N*-алфавітних систем**

N	K_N^v $(v=m)$									K_N
	v=1	v=2	v=3	v=4	v=5	v=6	v=7	v=8	v=9	
1	1									1
2	2	1								3
3	3	3	1							7
4	4	6	4	1						15
5	5	10	10	5	1					31
6	6	15	20	15	6	1				63
7	7	21	35	35	21	7	1			127
8	8	28	56	70	56	28	8	1		255
9	9	36	84	126	126	84	36	9	1	511

грамно-технічних комплексів широко розгалуженої в Україні автоматизованої інформаційної системи.

В межах цього програмно-технічного комплексу створено програмний модуль міжплатформеного обміну даними, що може функціонувати як на Web-сервері окремого комп'ютера, так і на Web-сервері, вмонтованому в СКБД Oracle. Модуль служить для завантаження файлів даних, що надходять з численних пунктів збору інформації до центральної бази даних (ОС Sun Solaris 9, СКБД Oracle), та для обміну даними з АІС, які мають інші платформи (наприклад, UNIX) та СКБД (приміром, MS SQL, Sybase Anywhere, Progress). При цьому можуть підтримуватися різні протоколи передачі даних (UUCP, FTP) та формати файлів (текстові, XML). Функції модуля розширені за рахунок додаткових функцій фільтрування даних при завантаженні їх у БД. Під час такого фільтрування здійснюється перевірка даних (реквізитів) на належність до певного алфавіту та приведення їх до алфавітної однорідності.

Програмний модуль написано мовою Java, яка, на наш погляд, найліпше підходить для створення розподілених додатків, орієнтованих на роботу з реляційними базами даних, різноманітними мережевими сервісами, підтримує сучасні технології розробки програмних комплексів, що працюють у середовищі Internet/intranet [8], а це є чи не найбільш важливою вимогою до програмних засобів модуля.

У роботі програмного модуля міжплатформеного обміну даними задіяні як внутрішні Java-класи (String, ResultSet, Prepared, Statement, Connection…), так і Java-класи, створені для виконання специфічних задач – завантаження інформації із різних файлів даних, файлів-квитанцій до центральної бази даних (Import21, Import22, ImportRI, ImportKVT…). Для реалізації алгоритму фільтрації даних додатково створено ще два Java-класи:

- langs служить для приведення символів даних до певного алфавіту і реалізує методи:
  - getLanquaqe – визначення мови даних;
  - intoUkrainian – заміни кодів символів на коди літер українського алфавіту;
  - intoRussian – заміни кодів символів на коди літер російського алфавіту;
  - intoLatin – заміни кодів символів на коди літер англійського алфавіту;
- SlashTokenizer, що служить для зчитування даних з файлів даних і реалізує методи:
  - removeillegal – видалення символів, заборонених для використання в базі даних;
  - LangTranslator – заміни літер кирилиці на літери англійського алфавіту, що збігаються за написанням.

Ці Java-класи викликаються в процесі завантаження інформації в БД, створюються їхні об'єкти, яким передається керування. За допомогою методів, перелі-

чених вище, проводиться посимвольний аналіз кожного реквізиту, зчитаного з файлу даних. Всі символи переводяться до верхнього регістру. Символи, які заборонені для використання в базі даних, вилучаються, а непередбачені замінюються на прогалики. Далі відбувається процес визначення алфавіту конкретного реквізиту, після чого згідно з алгоритмом проводяться заміни кодів символів, які не належать кодам даного алфавіту, на відповідні йому коди. Після завершення фільтрування реквізиту керування передається до класу, що викликав даний, який саме і здійснює процес завантаження відфільтрованої інформації до бази даних.

Під час функціонування модуля фільтруванню підлягає облікова інформація, сукупність якої однозначно ідентифікує особу або транспортний засіб. Такими відомостями можуть бути: прізвище, ім'я, по батькові, серія та номер паспортного документа, реєстраційний № автомобіля, рейс літака чи потяга.

На рис. 4 наведено блок-схему алгоритму приведення IP до алфавітної однорідності, за яким відбувається робота описаного програмного модуля, для розв'язання задачі усунення алфавітної неоднорідності записів IP у БД відповідно до табл. 3, де наведені літери та їхні коди (таблиця No 1251) для різних алфавітів, що збігаються за написанням.

На рис. 5 наведено блок-схему алгоритму визначення мови, за яким відбувається робота описаного програмного модуля, для розв'язання задачі розміщення IP у визначених для них полях БД.

В результаті нарощування функцій модуля за рахунок роботи фільтру інформація, що зберігається в БД, стає більш придатною для аналітичної обробки, час та ресурси, які витрачаються на пошукові операції в БД значно знижуються. Це служить більш ефективному використанню користувачами системи зібраної в БД інформації.

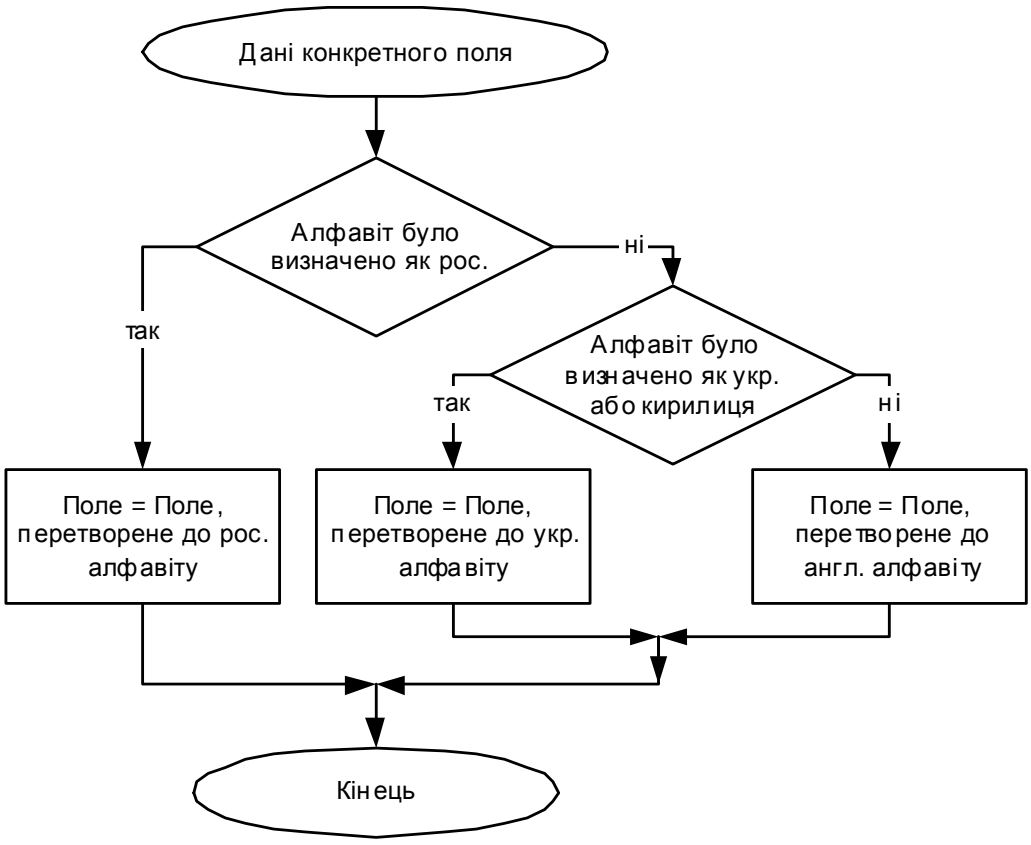

Рис. 4. Блок-схема алгоритму приведення IP до алфавітної однорідності

Таблиця 3

Літера	Кирилиця		Латиниця
	український алфавіт	Російський алфавіт	англійський алфавіт
I	178	–	73
A	192	192	65
B	194	194	66
E	197	197	69
K	202	202	75
M	204	204	77
H	205	205	72
O	206	206	79
P	208	208	80
C	209	209	67
T	210	210	84
X	213	213	88

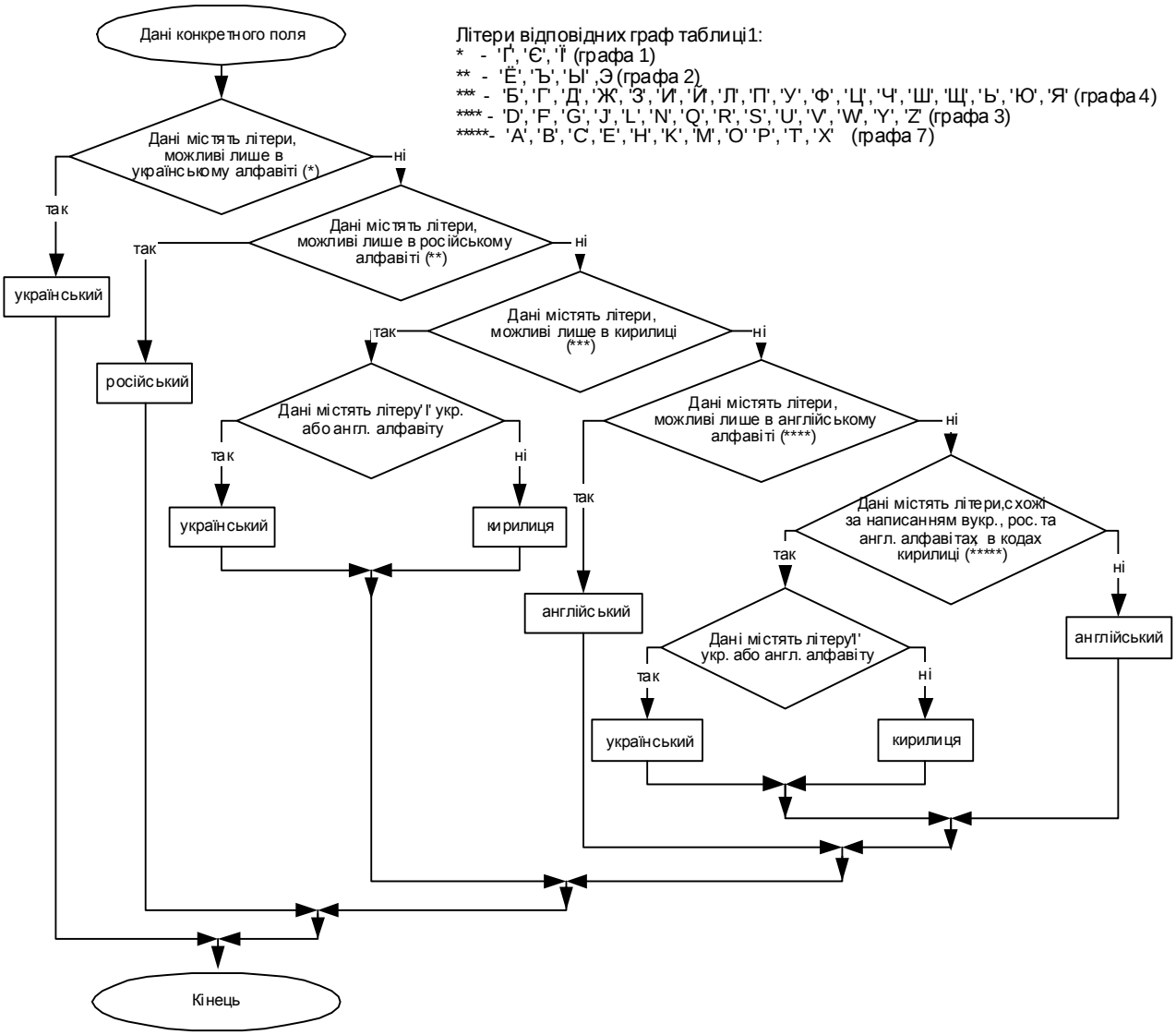

Рис. 5. Блок-схема алгоритма визначення мови

Висновки

Запропонований шлях визначення алфавітної однорідності або неоднорідності ІР дозволяє серед літер ІР, які належать одному алфавіту, гарантовано визначати літери, що хоча за написанням і збігаються, але за кодами (кодової сторінки) належать іншому алфавіту. Це дає змогу програмно поміняти коди таких літер, що неможна зробити при виборі інших шляхів, реквізиту на необхідні для вибраного алфавіту і, тим самим, збільшити алфавітну однорідність ІР у БД інформаційної системи, що у свою чергу знизить ймовірність помилок системи при інформаційно-пошукових операціях (відповідно до методів чіткого пошуку) за запитами по цих ІР і підвищить ефективність роботи самої системи.

1. *Словник іншомовних слів.* – Київ: Головна редакція УРЕ АН УРСР, 1974. – 776 с.
2. *Советский энциклопедический словарь.* – М.: Изд. «СЭ», 1989. – 1632 с.
3. *Першиков В.И., Савинков В.М.* Толковый словарь по информатике. – М.: Финансы и статистика. 1991. – 543 с.
4. ДСТУ 2226-93 (ГОСТ 34.003). Державний стандарт України. Автоматизовані системи. Терміни та визначення. – К.: Держстандарт України, 1994. – 5 с.
5. *Пентилюк М.І., Іващенко О.В.* Українська мова: Підручник-комплект. – К.: Ленвіт, 2001. – 352 с.
6. *Энциклопедия кибернетики* : В 2-х т. Т. 2. – Киев: Гл. ред. УСЭ, 1974. – 724 с.
7. *Бронштейн И.Н., Семендяев К.А.* Справочник по математике для инженеров и учащихся ВТУЗов. – Лейпциг: Изд. «Тойбнер», М.: Наука, 1980. – 976 с.
8. *Мак-Лахлин Б.* Java и XML: – Пер. с англ. – СПб: Символ-Плюс, 2002. – 544 с.

Отримано 29.03.04

Про авторів

Алексеєв Віктор Анатолійович,
канд. техн. наук, завідувач відділу

Ільїн Сергій Анатолійович,
провідний програміст

Терещенко Валерій Савелійович,
канд. техн. наук, старший науковий співробітник

Ференц Дар'я Антонівна,
провідний інженер-програміст

Місце роботи авторів:
Інститут програмних систем НАН України,
м. Київ, пр-т Академіка Глушкова, 40
тел.: (044) 526 4228, 526 6321
e-mail: alecseev@isofts.kiev.ua;
e-mail: terek@isofts.kiev.ua