

## МЕТОД ВИЯВЛЕННЯ СУПЕРЕЧНОСТЕЙ У БАЗАХ ЗНАНЬ СИСТЕМ ТЕХНІЧНОГО ДІАГНОСТУВАННЯ

*У статті розглядаються теоретичні основи процесу діагностування баз знань. Запропоновано метод виявлення суперечностей і розроблено його алгоритм.*

*Theoretical bases of process of diagnosticating of bases of knowledges are examined in the article. The method of exposure of contradictions is offered and his algorithm is developed.*

Ключові слова: база знань, критерій несуперечності, алгоритм виявлення суперечностей.

### Вступ

Збільшення рівня складності та зростання вимог щодо надійності сучасних технічних систем робить все більш актуальною проблему розробки відповідних засобів діагностування таких систем. Одним з шляхів вирішення цієї проблеми є використання інтелектуальних систем технічного діагностування [1, 2].

Якість функціонування інтелектуальних систем технічного діагностування (ІСТД) визначається кількістю діагностичної інформації у їхніх базах знань (БЗ) та стратегією їх використання. Найбільш розповсюдженими на сьогодні є системи технічного діагностування, бази знань яких побудовані з використанням продукційних правил [3]. Такі системи називають продукційними. Продукційні системи використовуються для діагностування складних технічних систем в медицині, аерокосмічній галузі, атомній енергетиці і т.і., що вимагає забезпечення високої якості рішень та надійності функціонування таких систем [4].

Принципові труднощі, які виникають при побудові ІСТД та їх баз знань, зумовлені рядом методологічних та технологічних проблем у галузі здобуття та опрацювання знань, зокрема це:

- слабкі теоретичні аспекти процесів здобуття знань;
- відсутність єдиного теоретичного базису процедури структурування знань;
- жорсткість моделей представлення знань;
- емпіричний вибір методів тестування систем, що базуються на знаннях;
- відсутність концептуальної цілісності та узгодженості між методами інженерії знань;
- відсутність техніко-економічних показників оцінки ефективності систем, що базуються на знаннях та інші.

Як наслідок, при наповненні баз знань експертні знання часто втрачаються або невірно трактуються, що призводить до неповноти опису предметної галузі, наявності суперечностей та протиріч у знаннях. При мовному спілкуванні інженер зі знань може отримати від експерта не більше ніж 24 % інформації, яку експерт хотів би усвідомити, сформулювати і передати.

Вище вказані труднощі спричиняють появу таких проблем, як надлишковість інформації, її суперечливість і неточність, що в сукупності знижує якість БЗ та ефективність використання ІСТД.

### Постановка задачі

Для підвищення ефективності ІСТД необхідно забезпечити відповідний рівень якості їх БЗ, зокрема розробити методи та засоби виявлення суперечностей у БЗ продукційного типу.

### Огляд відомих рішень

На даний час не існує єдиного підходу як до оцінки надійності баз знань, так і до оцінки цілісності, якості, повноти і несуперечності знань.

Оцінка якості експертної інформації є заключним етапом експертного опитування. Для оцінки якості експертної інформації пропонують наступну систему показників [5]:

- повнота (об'єм, потужність) отриманих знань ( $q_1$ );
- вірогідність інформації ( $q_2$ );
- точність інформації ( $q_3$ );
- несуперечність інформації ( $q_4$ );
- цінність інформації ( $q_5$ );
- оперативність отримання інформації ( $q_6$ );
- вартість отримання інформації ( $q_7$ );

Значення вказаних показників визначаються кількісним складом експертної групи  $e$  та  $k_i$  ступенем компетентності кожного  $i$ -го кандидата  $\lambda_i$ . Очевидно,  $\sum_{i=1}^k e_i = x$ , де  $e_i$  – кількість експертів  $i$ -го рівня компетентності.

На сьогодні пропонується ряд підходів, що базуються на методиках роботи з групами експертів котрі вносять знання в БЗ.

Задача експертної класифікації. Розглянемо формальну постановку задачі багатокритеріальної порядкової експертної класифікації:  $K = (K_1, K_2, K_N)$  – множина критеріїв, за якими оцінюється кожен об'єкт;  $S_q = \{k_1, k_2, K, k_q\}$  для кожного критерію  $K_q$  – множина оцінок, впорядкованих за відношенням характерності  $Q_q$ ;  $Y = S_1 \times S_2 \times \dots \times S_N$  – декартові похідні шкали критеріїв, що визначають простір станів

об'єктів, які класифікуються;  $Y$  с:  $Y$  – множина допустимих об'єктів;  $C = \{C_1, C_2, \dots, C_M\}$  – множина класів рішень, впорядкованих за відношенням характерності  $QC$ . Необхідно на основі експертних суджень побудувати відображення (класифікацію)  $F: Y \rightarrow \{Y\}$ ,  $i = 1..M$  (де  $Y_i$  – множина векторних оцінок, що належать класу  $C_i$ ,  $Y_i \cap Y_j = \emptyset \forall i \neq j$ ). Класифікація (відображення  $F$ ) повинна бути повною і несуперечливою.

Несуперечність характеризується [6]:

- відсутність в БЗ правил, що конфліктують;
- точністю розв'язання задач предметної галузі.

Конфлікт між правилами існує, якщо в процесі логічного виведення успішно активується більше ніж одне правило і в результаті їх активації вони суперечать одне одному. Такий стан є небажаним в БЗ.

Виділимо наступні форми конфліктів у БЗ: зациклювання, складність процесу логічного виведення [7]. *Конфлікт неперервності* виникає тоді, коли ліві частини правил ідентичні, а у правих частинах міститься суперечлива інформація. *Конфлікт ланцюжковості* схожий на конфлікт неперервності тим, що ліві частини початкових правил ідентичні, але в процесі виведення з'являються суперечливі стани (спочатку вони просто різні, а в результаті призводять до суперечностей).

*Конфлікт комплексності* виникає тоді, коли дві окремі умови з лівої частини правила задовольняються специфічними виключними станами робочої пам'яті, і в результаті виникає конфліктний результуючий стан пам'яті.

Джерелом конфлікту є недетермінізм в БЗ, тобто при тотожних вхідних даних не завжди отримуються однакові результати.

*Критерій*

Змінні:  $u, v, x, y$ ;

Константи  $a, b, c, k$ ;

*Прямий конфлікт:*

{R1 конфліктує з R2}

R1:  $P(x), P(y) \rightarrow Q(x, y)$

R2:  $P(x), P(y) \rightarrow \neg Q(x, y)$

*Ланцюжковий конфлікт:*

{конфлікт в ланцюжку суджень}

$P(x) \rightarrow R(y) \rightarrow \dots \rightarrow Q(x, y)$

$P(x) \rightarrow S(y) \rightarrow \dots \rightarrow \neg Q(x, y)$

*Комплексний конфлікт:*

{R3 та R4 приводять до конфлікту в робочій пам'яті WM}

R3:  $R(u), Q(v, w) \rightarrow Q(v, u), \neg R(u), R(w)$

R4:  $R(y), Q(x, k) \rightarrow \neg Q(x, k), Q(x, y), \neg R(y)$ .

R3 та R4 активізують:

WM:  $Q(b, k), Q(c, k), Q(a, b), R(a), R(c)$ .

Якщо R3 активізує підстановку  $c = u, a = v, b = w$ , то стан робочої пам'яті:

WM:  $Q(a, b), Q(a, c), Q(b, k), Q(c, k), R(a), R(b)$

Якщо R4 активізує підстановку  $b = x$  та  $c = y$ , то в WM робочій пам'яті:

WM:  $Q(b, c), Q(c, k), Q(a, b), R(a)$

Найбільш явний конфлікт полягає в тому, що у першому випадку робоча пам'ять WM містить  $R(b)$ , а в другому ні, при цьому предикат  $Q$  дає різні результати в обох WM.

Складність цього конфлікту полягає в тому, що звичайно результат не містить достатньої кількості деталей (предикатів) у правилі. Цей недолік не може бути виявлений звичайними структурними тестами.

#### Метод пошуку суперечностей у БЗ

Для виявлення суперечностей у БЗ пропонується метод, суть якого полягає у наступному: пошук суперечностей здійснюється почерговим порівнянням вхідних даних бази знань, у разі виявлення ідентичних значень порівнюються вихідні дані при цих значеннях і суперечлива інформація видається експерту для додаткового аналізу.

Для виявлення ланцюжкових конфліктів, вихідні дані одного продукційного правила порівнюються вхідними даними інших правил. Якщо виявлено зв'язок правил, знайдений вхідний елемент порівнюємо з відповідним йому вихідним і перевіряємо ці стрічки на несуперечність.

При відносно невеликій кількості правил (100-700), їх послідовний перебір дозволяє виявити суперечливі елементи за прийнятні терміни. Алгоритм пошуку суперечностей у БЗ наведено на рис. 2. Алгоритм передбачає читування рядків, що описують вхідні та вихідні дані різних правил та порівняння їх між собою.

На основі запропонованих методу та алгоритму було розроблено програмне забезпечення для виявлення суперечностей у БЗ. Приклад пошуку суперечностей у БЗ системи діагностування персональних комп'ютерів представлено на рис. 1. У результаті перевірки БЗ було виявлено 15 % правил, що суперечать одне одному. Такі правила вимагають подальшого уточнення експертом.

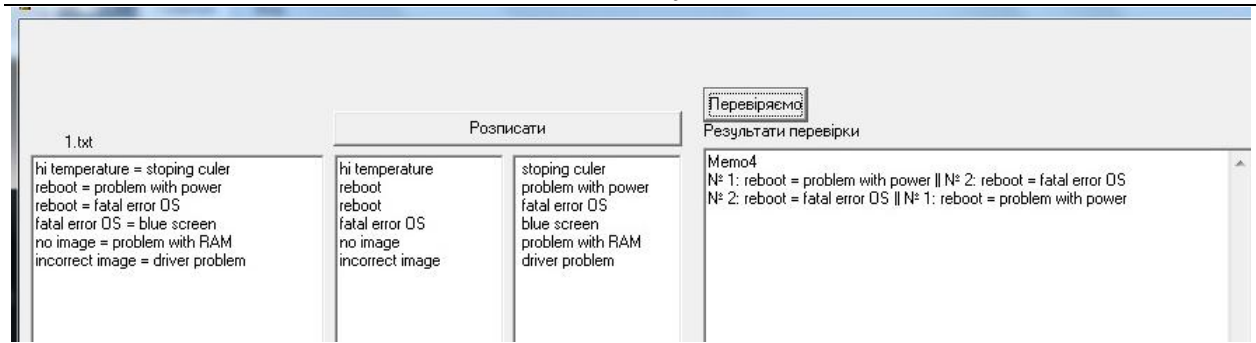


Рис. 1. Приклад результатів пошуку суперечливих правил у БЗ

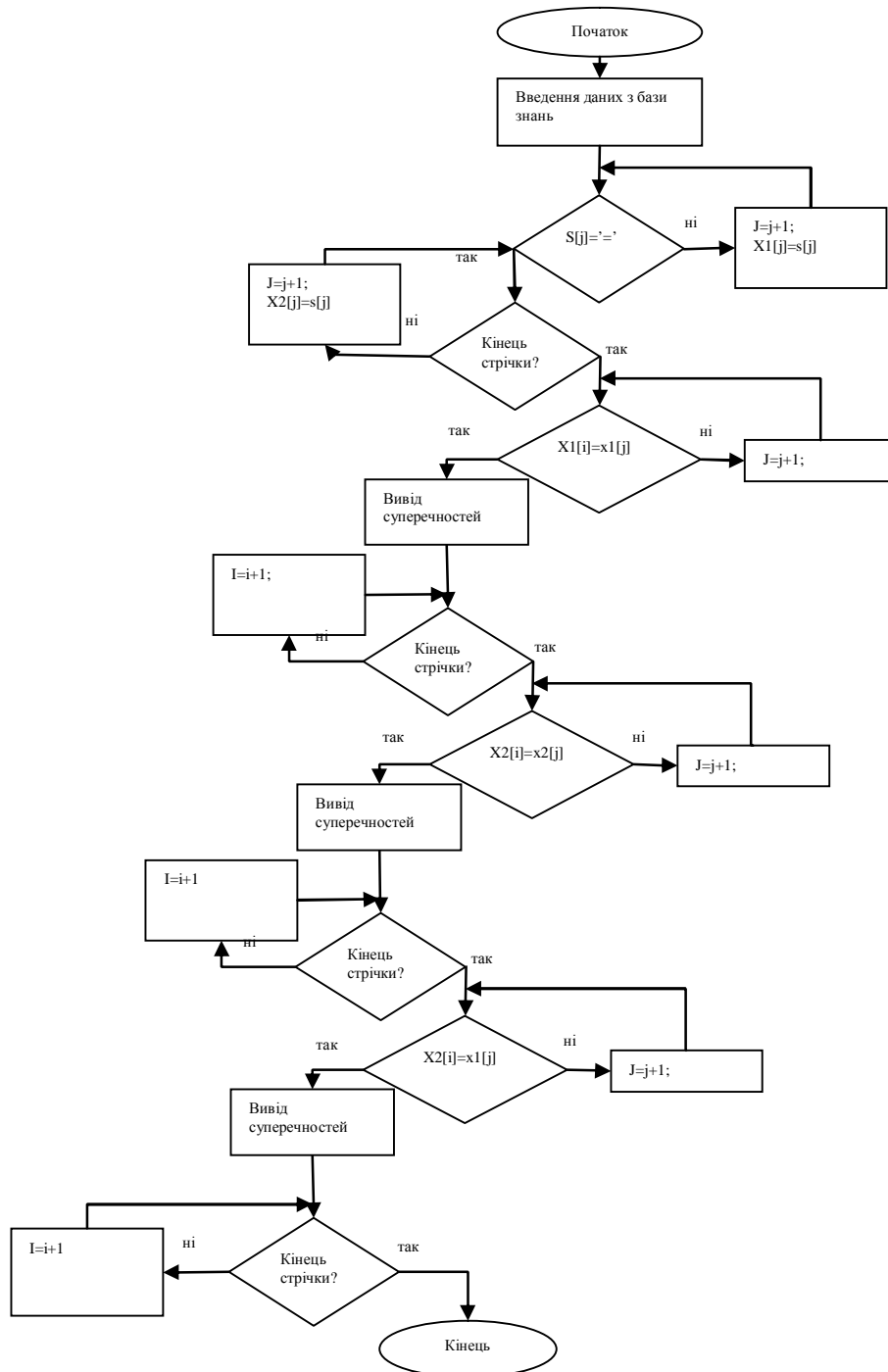


Рис. 2 Алгоритм пошуку суперечностей у БЗ

**Висновки**

Для підвищення якості БЗ ІСТД необхідно періодично перевіряти продукційні правила на суперечливість. Перевірка бази знань системи діагностування персональних комп'ютерів на наявність

суперечливих правил показала, що приблизно 5-15 % правил є суперечливими. Задіювання таких правил у процесі діагностування може давати суперечливі або невірні результати.

Запропонований метод пошуку суперечностей, що базується на порівнянні вхідних та вихідних даних різних правил, забезпечив виявлення прямих та ланцюжкових конфліктів у БЗ. Метод може використовуватись для БЗ невеликого об'єму (100-700) правил.

Подальшим напрямком досліджень є розроблення методів пошуку у БЗ конфліктів комплексного типу.

### Література

1. Hamscher W. C. Modeling Digital Circuits for Troubleshooting // Artificial Intelligence. – 1991. – Vol 51, № 1-3. – P. 223 – 271.
2. Поморова О. В. Априорна діагностична інформація в структурі нейромережних експертів ідентифікації стану компонентів комп'ютерних систем // Радіоелектронні і комп'ютерні системи. – 2007. – № 8. – С.145 – 151.
3. Попов Э. В. Экспертные системы: Решение неформализованных задач в диалоге с ЭВМ. – М.: Наука, 1987. – 288 с.
4. Джексон П. Введение в экспертные системы. – М.: СПб., Киев: "Вильямс", 2001. – 624 с.
5. Гаврилова Т.А., Хорошевский В.В. Базы знаний интеллектуальных систем – СПб: Питер, 2001. – 384с.
6. Герасимов Б.М., Субач І.Ю. Показники якості інформаційного забезпечення та їх вплив на ефективність застосування ІСППР. Вісник Київського національного університету ім. Т. Шевченка; Військово-Спец. науки, 2008. – № 2, С. 18-25.
7. Поморова О.В., Гнатчук Є.Г. Виявлення суперечливості правил нечітких баз знань інтелектуальних систем технічного діагностування.

Надійшла 23.5.2010 р.

УДК 004.272.2

Д.М. МЕДЗАТИЙ, А.О. МАМЧУР, Л.А. МАТВІЙЧУК, В.В. ГОНЧАРУК  
Хмельницький національний університет

## АНАЛІЗ ТА СИНТЕЗ ТОПОЛОГІЙ РОЗПОДІЛЕНИХ ОБЧИСЛЕНЬ

*В статті розглянуто основні принципи та особливості використання топологічного аналізу для оптимізації паралельного обчислювального процесу. Проведено аналіз методів представлення топологій обчислювального процесу. Наведено приклад реалізації запропонованого підходу.*

*The article reviews the basic principles and specific use of topological analysis for optimizing parallel computation process. The analysis methods of presentation of topology computation process. Implemented the proposed approach.*

Ключові слова: паралельні обчислення, розподілені обчислення, топологія обчислювального процесу.

### Вступ

На сьогодні розподілені обчислення є одним з ефективних засобів розв'язання трудомістких задач з використанням двох або більше комп'ютерів, об'єднаних в мережу. Доступність та економічна доцільність цього підходу (особливо для систем типу Beowulf) робить його привабливим для застосування у навчальних закладах, наукових лабораторіях та інших установах, де є потреба у розв'язання трудомістких задач. Однак, розроблення програмного забезпечення (ПЗ), яке використовує розподілені обчислення для розв'язання поставлених задач, не є простим завданням. Основними проблемами розробки такого ПЗ є складність контролю виконання та складність оптимального розподілення задач між вузлами мережі, а також синхронізація даних.

З ускладненням обчислювального процесу зростає складність контролю над ним, збільшується ймовірність виникнення помилок та зменшується продуктивність системи. Наприклад, впровадження нового проміжного рівня обчислення задачі може призвести до повної зміни обчислювального процесу, проконтролювати який досить складно, а в деяких випадках і зовсім неможливо.

### Аналіз досліджень та публікацій

Відомі підходи до розв'язання задачі проектування паралельних алгоритмів і програмного забезпечення зорієнтовані на розроблення систем шляхом аналізу поставленої задачі та можливих шляхів її розв'язання [1, 2]. Широко використовується масштабування підзадач та їх розподіл між процесорами [1, 3, 4]. Разом з тим, на нашу думку, у дослідженнях недостатньо висвітлені питання автоматизації процесу розроблення програмних засобів орієнтованих на виконання у розподілених системах. Задачі аналізу та синтезу «паралельних» алгоритмів та програмного забезпечення цілком покладаються на програміста. Таке положення ставить під сумнів ефективність та оптимальність отриманих програмних засобів. Отже, задача розроблення підходів щодо аналізу та оптимізації топологій систем розподілених обчислень, які придатні до автоматизації, є актуальною.

Одним з варіантів автоматизації процесу аналізу та синтезу топологій систем розподілених