УДК 004:891 DOI:10.31673/2412-4338.2021.020414

Павленко М.А., Осієвський С.В. *Харківський національний університет Повітряних Сил імені Івана Кожедуба, Харків*
Опенько П.В. *Національний університет оборони України імені Івана Черняховського, Київ*
Олімпієва Ю.І. *Державний університет телекомунікацій, Київ*

ПРОЦЕДУРА ЗНЕВАДЖЕННЯ ПОМИЛОК КЛАСУ РЕПЛІКА В ПРОДУКЦІЙНИХ БАЗАХ ЗНАНЬ

Анотація: *Розглянуто завдання виявлення помилок в продукційних базах знань знання – орієнтованих інформаційних систем, що виникають на етапі формування бази знань експертами. Визначено, що такі помилки пов'язані з суперечливістю думок експертів та/або обмеженістю (недосконалістю) опису предметної області. Проаналізовано підходи щодо їх зневадження. Показані шляхи вдосконалення існуючих підходів щодо зневадження статичних помилок класу "репліка" . Показано можливі шляхи застосування отриманих рішень для зневадження помилок "суперечливість", "надмірність", "неповнота". Запропоновані рішення щодо розширення формалізованого визначення статичної помилки продукційной бази знань з урахуванням вимог до точності подання інформації. Розглянуті питання впливу помилок класу "репліка" на результати виведення за правилами продукційної бази знань. Доведено можливість застосування методів теорії графів для вирішення завдання зневадження помилок класу "репліка" . Розроблені алгоритмічні структури виявлення та зневадження помилок зазначеного класу, які дозволяють , на відміну від існуючих рішень виявляти дублюючі вершини на кожному ранзі графа до якого зведена продукційна база знань. Розроблено програмну реалізацію виявлення та зневадження статичних помилок неповного, часткового та повного дублювання. За рахунок застосування рекурсії знижено вимоги щодо підготовки масиву вхідних даних до обробки. Отримані рішення відповідають вимогам ДСТУ ISO/IEC 9126, ДСТУ ISO/IEC 14598 та враховують вимоги серії стандартів Software Quality Requirements and Evaluation як значення вершин графа дерева подій. В процесі рішення завдання врахована специфіка функціонування ЛМС, зокрема можливість формалізації різних аспектів знань (алетичних, дисизіональних, каузальних, діонтичних) та забезпечення заданого рівня оперативності пошуку рішень.*

Ключові слова*: граф, модель, методика, продукційна модель, база знань.*

Pavlenko M. A, Osiievskyi S.V. *Ivan Kozhedub Kharkiv National Air Force University, Kharkiv*
Open'ko P. V. *The National Defence University of Ukraine named after Ivan Chernyakhovsky, Kyiv*
Olimpiyeva Ju. I. *State University of Telecommunications, Kyiv*

DEBUGGING PROCEDURE OF THE REPLICA CLASS ERRORS IN THE PRODUCTION KNOWLEDGE BASES

Abstract: *The problem of detecting errors in the production knowledge bases of knowledge - oriented information systems that arise at the stage of formation of the knowledge base by experts is considered. It was determined that such errors related to the conflicting opinions of experts and/or limited (imperfect) description of the subject area. There was analyzed approaches for their debugging. Ways to improve existing approaches for debugging static replica errors was shown. Possible ways of applying the obtained solutions to debug errors "contradiction", "redundancy", "incompleteness" was shown. Solutions for expanding the formalized definition of static error of the production network, considering the requirements for the accuracy of information was suggested. The issues of the influence of errors of the "replica" class on the results of derivation according to the rules of the production knowledge base was considered. The possibility of applying the methods of graph theory to solve the problem of error reduction of the "replica" class is proved. An algorithmic structure for detecting and debugging errors of this class has been developed. It allows, in contrast to existing solutions, to detect duplicate vertices at each rank of the graph to which the production knowledge base is reduced. Software implementation for detection and debugging of static errors of incomplete, partial, and complete duplication was developed, using recursion, which allows to reduce the requirements for preparing an array of input data for processing. The obtained solutions meet the requirements of DSTU ISO /*

IEC 9126, DSTU ISO / IEC 14598 and consider the requirements of the series of standards Software Quality Requirements and Evaluation as the values of the vertices of the graph of the event tree. During problem solving process specifics of the LMS operation are considered, first of all formalizing possibilities in the various aspects of knowledge (aletic, dissociative, causal, diontic) and ensuring a given level of efficiency in finding solutions.

Keywords: *graph, model, methodology, production model, knowledge base*

Павленко М. А., Осиевский С. В. *Харьковский национальный университет Воздушных Сил имени Ивана Кожедуба, Харьков*
Опенько П.В. *Национальный университет обороны Украины имени Ивана Черняховского, Киев*
Олимпиева Ю. И. *Государственный университет телекоммуникаций, Киев*

ПРОЦЕДУРА ОТЛАДКИ ОШИБОК КЛАССА РЕПЛИКА В ПРОДУКЦИОННЫХ БАЗАХ ЗНАНИЙ

Аннотация: *Рассмотрены задачи обнаружения ошибок в продукционных базах знаний знания - ориентированных информационных систем, возникающих на этапе формирования базы знаний экспертами. Определено, что такие ошибки связаны с противоречивостью мнений экспертов и/или ограниченностью (несовершенством) описания предметной области. Проанализированы подходы к их отладке. Показаны пути совершенствования существующих подходов к отладке статических ошибок класса "реплика". Показаны возможные пути применения полученных решений для устранения ошибок "противоречивость", "избыточность", "неполнота". Предложены решения по расширению содержания определения статической ошибки продукционной базы знаний с учетом требований к точности представления информации. Рассмотрены вопросы влияния ошибок класса "реплика" на результаты вывода по правилам продукционной базы знаний. Доказана возможность применения методов теории графов для решения задачи отладки ошибок класса "реплика". Разработаны алгоритмические структуры выявления и отладки ошибок указанного класса, которые позволяют, в отличие от существующих решений, выявлять дублирующие вершины на каждом ранге графа к которому сведена продукционная база знаний. Разработана программнаю реализация выявления и отладки статических ошибок неполного, частичного и полного дублирования. За счет применения рекурсии снижены требования по подготовке массива входных данных к обработке. Полученные решения отвечают требованиям ДСТУ ISO / IEC 9126, ДСТУ ISO / IEC 14598 и учитывают требования серии стандартов Software Quality Requirements and Evaluation как значение вершин графа дерева событий. В процессе решения задачи учтена специфика функционирования ЧМС, в частности, возможность формализации различных аспектов знаний (алетических, диссизиональных, казуальных, диантических) и обеспечения заданного уровня оперативности поиска решений.*

Ключевые слова: *граф, модель, методика, продукционная модель, база знаний*

1. Вступ

Всебічне нарощування можливостей автоматизації прийняття рішення в слабоформалізуємих областях, за рахунок використання експертних знань, призводить до зростання числа помилок в програмному забезпеченні знання-орієнтованих інформаційних систем (ПЗ ЗОІС). В [1], на підставі детального аналізу, зроблено висновок про те, що проблема виявлення та усунення помилок загострюється в міру збільшення складності завдань, що вирішуються програмним забезпеченням, і загрожує катастрофами в системах, що виконують критичні функції управління великими, дорогими і особливо важливими об'єктами або процесами. Зазначені аспекти вимагають розробки відповідних алгоритмів, моделей і методів які забезпечать необхідну якість програмного забезпечення в цілому та програмного забезпечення ЗОІС, зокрема. Одним з основних засобів, за допомогою якого досягається висока надійність правильного функціонування та якість ПЗ ЗОІС, є діагностування програмного забезпечення (ПЗ) на різних етапах його життєвого циклу, зокрема, на етапах проектування, розробки та експлуатації. Важливою складовою діагностування ПЗ ЗОІС є зневадження та тестування. Їх роль зростає з введенням припущення в тому, що ПЗ ЗОІС є досить складним і не може бути бездефектним. Саме тому виявлення недоліків методів та

засобів зневадження та тестування ПЗ ЗОІС, у тому числі ідентифікація прихованих помилок програмних додатків, є актуальною задачею розвитку та вдосконалення системи підтримки процесів розробки ПЗ ЗОІС.

2. Аналіз досліджень і публікацій

Можливі шляхи вирішення завдання розробки методів зневадження баз знань досить детально викладені в працях таких вчених, як Ліпаєв В.В, Зикова С.А., Наріньяні А., Zadeh L., Tsukamoto Y., Miller S. [2-6] та ін. З аналізу цих праць та публікацій зрозуміло, що найбільшого успіху досягнуто в питанні розвитку методів статичного зневадження, тобто тих методів, що не вимагають запуску експертної системи на виконання. Однак на сьогоднішній день не існує єдиного підходу до формалізації так званих структурних помилок, що виявляються методами статичного зневадження: надмірність, неповнота, суперечливість. Саме тому статично коректні бази знань не гарантують якості прийнятих рішень за рахунок помилок в самих знаннях, часто пов'язаних зі складністю окремої предметної області.

3. Ціль (мета) дослідження. Зважаючи на результати аналізу та наявні рішення в галузі проектування ЗОІС визначимо, що метою даної публікації є розробка механізмів виявлення дублікатів БЗ моделі підтримки процесів розробки інтелектуальних СППР (ІСППР), що дозволить усунути зазначені проблеми.

4. Результати дослідження

4.1 Загальний підхід до розробки методології зневадження баз знань

У відповідності до [8], зневадження – методичний процес пошуку та зменшення числа помилок чи дефектів у комп'ютерній програмі або електронному обладнанні з метою отримання очікуваної поведінки. Зневадження, як правило, ускладнюється, коли різні підсистеми міцно пов'язані між собою, оскільки зміни в одній частині можуть викликати помилки в іншій. Тобто, метою процесу зневадження баз знань є виявлення та усунення максимального числа помилок у ПЗ ЗОІС.

Помилки бази знань прийнято розділяти на наступні групи:

- помилки, що пов'язані з порушенням внутрішньої структури бази знань;
- помилки, що пов'язані із зовнішніми протиріччями бази знань.

Для виявлення першої групи помилок базу знань подають у вигляді графової моделі та формалізують всі можливі помилки, що пов'язані з порушенням внутрішньої структури самої бази знань. До таких помилок відносяться неповнота, надмірність, внутрішня суперечливість. Наступним етапом є виявлення помилок даної групи і виправлення бази знань, приводячи її до стану статичної коректності. Оскільки термін статичної коректності найбільш повно відображає сутність вирішення поставленого завдання, далі в роботі він буде використовуватись з врахуванням його трактування у [8, 9].

Відзначимо, що встановлення факту статичної коректності бази знань не гарантує того, що статично коректна база знань не робить невірні висновки через можливі помилки в самих правилах, причинами яких є зовнішня суперечливість самої предметної області і помилки в самих правилах. Введемо визначення зовні суперечливої бази знань у вузькому сенсі слова.

Слід зазначити, що більшість предметних областей, за своєю природою, є зовні суперечливими та містять помилки, пов'язані з обмеженнями в предметній області або з наявністю фактів критичного поєднання подій, що призводять до помилок в прийнятті рішення. Зрозуміло, що вищезазначені помилки не пов'язані зі структурою знань та не можуть виявлятися графовими методами та методами статичного аналізу. З чого слідує висновок, що завдання виявлення таких помилок може бути реалізоване лише за рахунок тестування самої бази знань. І, як наслідок, має місце твердження, що статичне зневадження, в ході якого виявляються та усуваються структурні помилки бази знань та внутрішня суперечливість, є необхідною, але не достатньою умовою забезпечення якості БЗ.

Як було зазначено вище, для виявлення помилок, що пов'язані з порушенням внутрішньої структури бази знань, її необхідно привести до графової моделі. У нашому випадку (оскільки мова йде про статичний аналіз бази знань) передбачається, що структура БЗ відома. Цей факт дозволяє, побудувати графову модель (структури ТА/АБО) і виконати її

наступний аналіз. На цьому етапі варто ввести обмеження, що полягає в наступному: зазначені рішення являються дієвим механізмом для виявлення помилок, що пов'язані з порушенням внутрішньої структури та зовнішніми протиріччями бази знань лише для продукційних баз знань. Це пов’язано з тим, що для нейронної мережі структура знань, отримана нейронною мережею в ході навчання, є неявна; вона сформована в множині вагових коефіцієнтів зв'язків між нейронами, які не піддаються інтерпретації. Крім того, сама збалансована структура нейронної мережі не дозволяє з впевненістю говорити про наявність в ній таких класів помилок як «неповнота», «надмірність» та «суперечливість».

4.2 Моделі помилок в продукційних базах знань

Для продукційних баз знань (ПБЗ) ЗОІС в літературі [10 - 16] виділяють наступні основні класи помилок: неповнота, надмірність, суперечливість. Однак, незважаючи на наявні роботи в даній області, до цих пір відсутній системний підхід щодо формалізації основних типів структурних помилок. В роботі в окремий клас “Репліки” пропонується винести помилки типу “Дублікати” та саме їх розглянути в даній роботі. Основні класи структурних помилок ПБЗ представлені на рисунку 1, окремо виділено клас “Репліки”, що пропонується до розгляду.

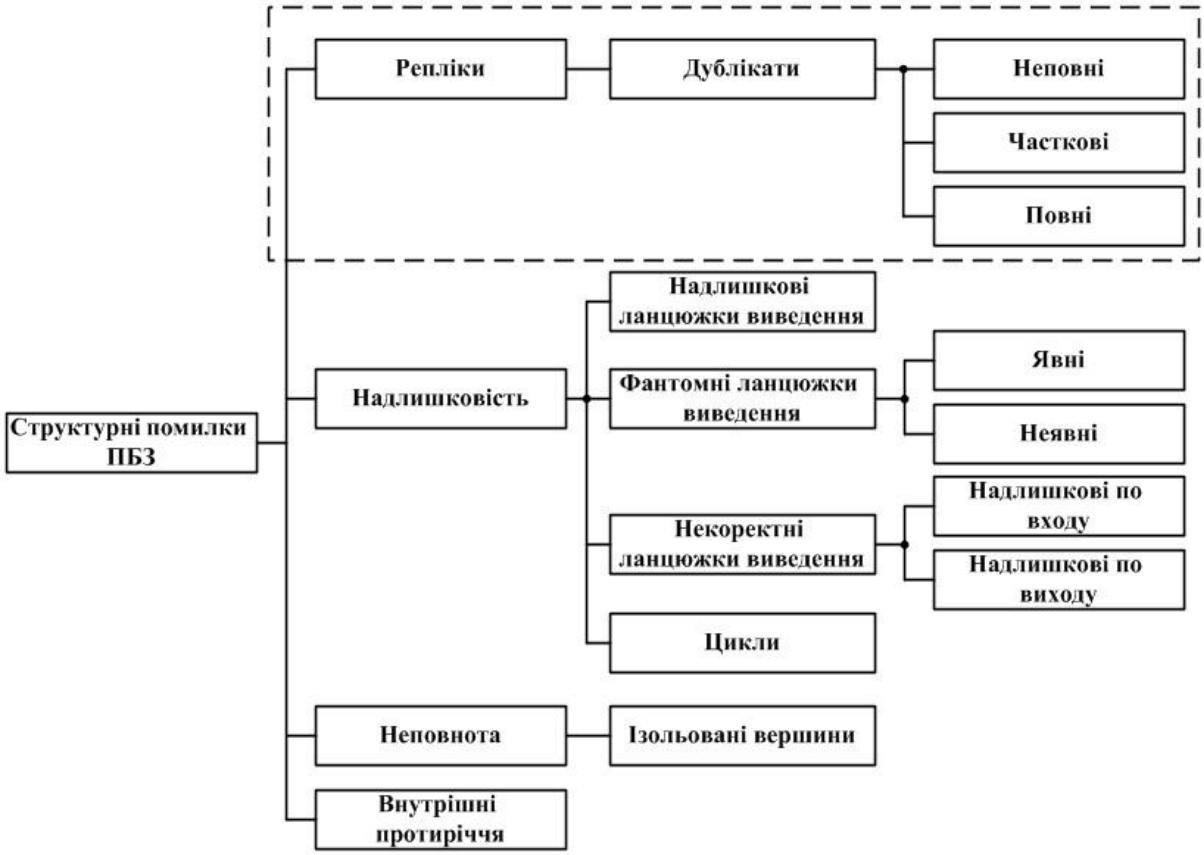

Рис.1. Класи структурних помилок продукційної бази знань

Формалізуємо структурні помилки продукційної бази знань з точки зору теорії графів.

З метою наочної візуалізації безпосередніх теоретичних положень, в якості структури продукційної бази знань, що розглядається, використаємо гібридну мережну модель знань про процес відбору джерел нарядів вогневих засобів Повітряних Сил, наведену в [17, стор 34] та з врахуванням змістової частини [18].

Формалізований опис фрагменту вищезазначеної моделі в термінах ПБЗ матиме наступний вигляд:

$$r_1 : якщо\ s_1\ та\ s_2, то\ f_1;$$
$$r_1 : якщо\ s_2\ та\ s_3, то\ f_2;$$
$$r_3 : якщо\ s_3\ та\ s_4\ , то\ f_3;$$

$$r_4 : якщо\ f_1, то\ g_1;$$
$$r_5 : якщо\ f_2\ та\ f_3, то\ g_2;$$
$$\text{де } s_1, s_2, s_3, s_4 \in S, F;$$
$$f_1,\ f_2,\ f_3 \in F;$$
$$r_1, r_2, r_3, r_4, r_5 \in R;$$
$$g_1, g_2 \in G,$$

де F – множина фактів ЗОІС, що об'єднує вхідні, проміжні та термінальні факти; R – множина правил, що мають вигляд $R : if\ (\ f_{i_1}\ \&\ f_{i_2}\ \&...\&\ f_{i_k}\ )\ then\ \ f_{i_{(i+1)}}$.

Даному опису відповідає ТА/АБО G_s граф, наведений на рисунку 2.

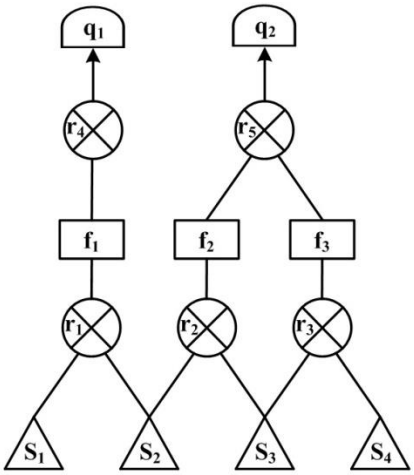

Рис.2. Структура ТА/АБО G_p графа продукційної бази знань (фрагмент)

Сформовані в структурі графу зв'язки будемо називати ланцюжками виведення. Тобто, в загальному випадку, ланцюжок виведення l_i – послідовність правил $(r_{l_i 1}, r_{l_i k} r_{l_i 2},...,r_{l_i k})$ така, що $\forall r_{l_i k}, r_{l_i k+1}, q_{r_{l_i k}} \in D_{r_{l_i k+1}}$ при $k = 2,...,n-1$, де D_r множина фактів, що належать умовній частини правила r. Тоді для графа, наведеного на рисунку 2, L є ланцюжком виведення, де $L = \{l_1, l_2, l_3, l_4, l_5, l_6, l_7, l_8, \}$, де $l_1 = (r_1)$, $l_2 = (r_2)$, $l_3 = (r_3)$, $l_4 = (r_4)$, $l_5 = (r_5)$, $l_6 = (r_1, r_4)$, $l_7 = (r_2, r_5)$, $l_8 = (r_3, r_5)$. Логічно, що для ланцюжку виведення l_i початок ланцюжка виведення l_i виду $(r_{l_i 1}, r_{l_i k} r_{l_i 2},...,r_{l_i k})$ є множиною фактів в умові виконання першого правила, $D_{l_i} = D_{r_{l_i 1}}$, а кінець ланцюжка виведення l_i є результатом виконання останнього правила, $G_{l_i} = G_{r_{l_i 1}}$.

Наведений опис відображає загальний підхід до формалізації структурних помилок продукційної бази знань з точки зору теорії графів та буде використаний в подальшому викладенні матеріалу.

4.3 Формалізація структурних помилок класу "Репліка"

У відповідності до класифікації структурних помилок клас "Репліка" містить три складових: неповні дублікати, часткові дублікати та повні дублікати. Формалізуємо кожну із складових зазначеного класу та наведемо алгоритми їх виявлення.

Правила r_i і r_j називаються дублікатами, якщо вони мають хоча б один загальний факт в умовах виконання та один і той же результат виконання. Тобто формально задача пошуку неповних дублікатів для графа G_s полягає в тому, що для заданого графа ЗОІС системи G_p необхідно знайти пари вершин з підмножини V_1 (підмножина вершин), з'єднаних двома (або більше) орієнтованими ланцюгами рангу 2. На рисунку 3 наведено приклад неповних дублікатів.

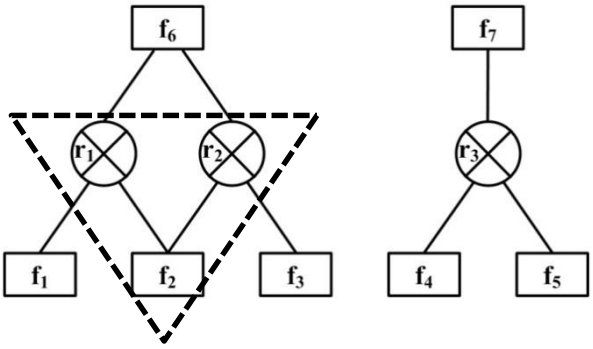

Рис.3. Приклад неповних дублікатів (граф G_s)

Формалізований опис наведеного на рисунку 3 графа G_s має наступний вигляд:

$$r_1 : якщо\ f_1\ та\ f_2, то\ f_6$$
$$r_2 : якщо\ f_2\ та\ f_3, то\ f_6$$
$$r_3 : якщо\ f_4\ та\ f_5, то\ f_7$$
$$D_{l_1} \cap D_{l_2} = f_2$$

Заходи для виправлення помилки типу “неповні дублікати” для кожного конкретного випадку визначаються експертом та залежать від глибини вкладень дублікатів у загальній системі правил ПБЗ.

Завдання пошуку неповних дублікатів вирішується за допомогою модифікованої процедури обходу в ширину для заданого графа та початкових вершин з множини V_1. Пошук в ширину здійснюється не до вичерпання всіх досяжних вершин графа, а лише до вершин, що знаходяться на другому ранзі від початкової. Структура модифікованого алгоритму процедури пошуку неповних дублікатів для графа G_s, що визначає помилку типу **“**неповні дублікати**”** для будь-якої вершини f представлений на рисунку 4. Дамо коротке пояснення наведеній алгоритмічній структурі. На підготовчому етапі формується масив вершин для яких існує двонаправлений зв'язок з цільовою вершиною vertex та виконується їх запис до черги.

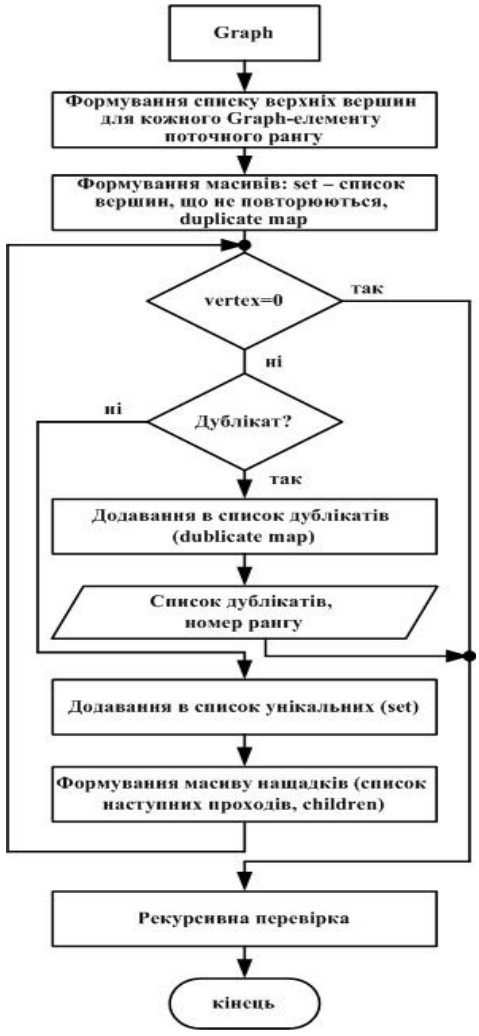

Рис. 4. Структура модифікованого алгоритму процедури пошуку неповних дублікатів для графа G_s

Для всіх вершин в черзі (ранг яких становить 1) виконується операція видалення з черги та запам’ятовування поточної позиції двостороннього зв’язку, виконується перегляд суміжних з поточною вершин, якщо виникає спроба повторного додавання вершини до черги (вершина не є новою і вже переглядалася для існуючого правила)

вершина помічається як дублікат та фіксується її ранг..

Складність алгоритму пошуку дублікатів для одного факту не перевищує складності обходу в ширину в гіршому випадку $O(V+E)$ [19], але оскільки алгоритм повторюється для кожного рангу БЗ ЗОІС, то, складність пошуку неповних дублікатів становитиме $O((V+E)\times Level)$, де $Level$ – кількість рангів БЗ ЗОІС. На рисунку 5 наведено фрагмент лістингу процедури пошуку неповних дублікатів на всіх рангах графу G_s.

```
import java.util.*;
public class Vertex {
   private final int level;
   private final String name;
   private final List relations = new ArrayList();
   public Vertex(int level, String name) {
      this.level = level;
      this.name = name;}
   public void addRelation(Vertex child) {
      relations.add(child);}
   public List<Vertex> getChildren() {
      return relations;}
   public static HashMap<Integer, List<Vertex>> findDuplicates(List<Vertex> graph) {
      List<Vertex> nextLevelVertexes = new ArrayList();
      for (Vertex vertex :
            graph) {//пошук усіх вершин вищого рангу
         nextLevelVertexes.addAll(vertex.getChildren());}
      Set<Vertex> listOfCheckedVertexesVisuotDuplicates = new HashSet<>();//унікальні вершини
      HashMap<Integer, List<Vertex>> duplicatesMap = new HashMap<>();//integer - vertex level,
список дублікатів
…
            if (!duplicatesMap.containsKey(vertex.level)) {
               duplicatesMap.put(vertex.level, new ArrayList<Vertex>()); }
            // додавання елементу в список дублікатів по рангу
            duplicatesMap.get(vertex.level).add(vertex);
         } else {
            // якщо не дублікат – збереження в масив унікальних
            listOfCheckedVertexesVisuotDuplicates.add(vertex);
            //якщо існують нижчі рівні, формування їх списку для рекурсивного обчислення
дублікатів
            if (vertex.getChildren() != null && !vertex.getChildren().isEmpty()) {
               childOfChildList.add(vertex); }
         }
      }
…
   @Override
   public boolean equals(Object o) {
      if (this == o) return true;
      if (o == null || getClass() != o.getClass()) return false;
      Vertex vertex = (Vertex) o;
      return level == vertex.level && Objects.equals(name, vertex.name) && Objects.equals(relations,
vertex.relations);}
   @Override
   public int hashCode() {
      return Objects.hash(level, name, relations);}
}
```

Рис. 5. Фрагмент лістингу процедури пошуку неповних дублікатів за всіма рангами графу

Наведена процедура пошуку неповних дублікатів передбачає роботу з невпорядкованими списками, що суттєво зменшує час виконання операцій підготовки масиву

даних.

Наявність часткових дублікатів у ПБЗ ЗОІС визначається ситуацією коли для правил r_i і r_j має місце відношення $D_{r_i} \subset D_{r_j}$, при цьому $D_{r_i} \neq D_{r_j}$ та, відповідно $q_{r_i} \subset q_{r_j}$. Іншими словами, множина фактів, що належать умовній частині правила r_i являється складовою множини фактів, що належать умовній частині правила r_j. Наприклад, правила r_1 та r_2 на рисунку 6 є частковими дублікатами

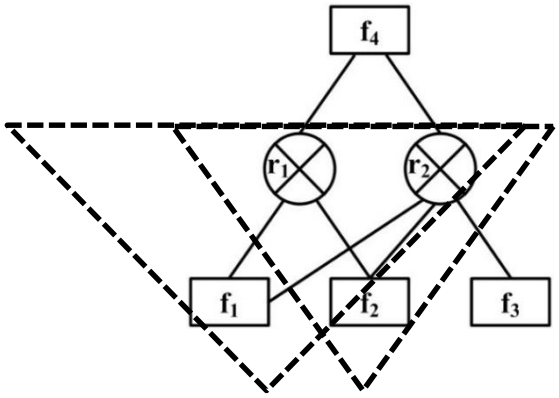

Рис.6. Приклад часткових дублікатів (граф G_s)

Формалізований опис наведеного на рисунку 6 графа G_s має наступний вигляд:

$$r_1 : якщо\ f_1\ та\ f_2, то\ f_4;$$

$$r_1 : якщо\ f_3\ та\ f_2\ та\ f_1\ то\ f_4;$$

$$D_{r_1} = \{f_1, f_2\};$$

$$D_{r_2} = \{f_1, f_2, f_3\}$$

$$D_{r_1} \subset D_{r_2};$$

$$q_{r_1} = f_4;$$

$$q_{r_2} = f_4.$$

Наявність часткового дубліката в формальному описі свідчить про те, що в графі G_s існують дві вершини r_1 і r_2 такі, для яких одночасно виконуються наступні умови:

$\exists f \in V_1(r_1, f) \in E \,\&\, V_1(r_1, f) \in E$ (факт приналежності частини правила виведення);

$\forall f \in V_1(f, r_1) \in E \Rightarrow V_1(f, r_2) \in E$ (включення частини умови правила виведення r_1 в частину умови правила виведення r_2);

$\exists f \in V_1(r_1, f) \in E \,\&\, V_1(r_2, f) \notin E$ (суворе включення частин умови правил r_1 та r_2).

Алгоритмічне рішення зазначеного завдання ідентичне рішенню завдання пошуку неповних дублікатів з умовою зміни рангів цільових вершин на непарні. Оцінка складності алгоритму пошуку часткових дублікатів залежить від характеристик структур даних, що представляють граф G_s. І єдине на що може вплинути розробник це застосування рішення union-find для збереження структур даних, оскільки застосування такого рішення вимагає для представлення всієї структури графа G_s обсягу пам'яті, що пропорційний кількості дуг зазначеного графа.

Задача пошуку повних дублікатів для графа G_s полягає в тому, що для заданого графа експертної системи G_p необхідно знайти пари вершин з підмножини V_1, такі, що множина їх вихідних вершин єдина, а множина вхідних вершин задовольняє відношенню суворого включення. Правила r_1 та r_2 на рисунку 7 є повними дублікатами. Формалізований опис

наведеного на рисунку 7 графа G_s має наступний вигляд:

r_1 : *якщо* f_1 *та* f_2 *та* f_3 *та* f_4, *то* f_5;

r_2 : *якщо* f_1 *та* f_2 *та* f_3 *та* f_4, *то* f_6

r_3 : *якщо* f_3 *та* f_4, *то* f_6.

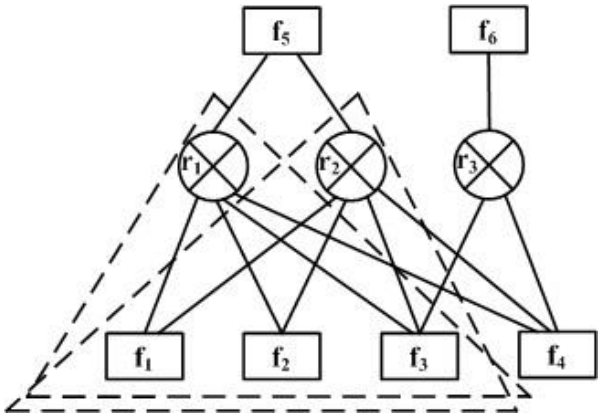

Рис.7. Приклад повних дублікатів (граф G_s)

Вирішення проблеми зневадження повних дублікатів вбачається у видаленні усіх дублікатів крім одного, наявність повних дублікатів свідчить про абсолютне співпадіння двох або більше правил в ПБЗ. З формалізованого опису слідує наявність співпадіння частин умов для правил виведення $\forall f \in V_1(f, r_1) \in E \,\&\, V_1(f, r_2) \in E \,\&\, V_1(f, r_3) \in E \,\&\, V_1(f, r_4) \in E$.

Алгоритмічна структура пошуку повних дублікатів ідентична алгоритмічній структурі процедури пошуку неповних дублікатів та відрізняється лише вибором і парних і непарних рангів розміщення цільових вершин.

Зазначимо, що розроблена алгоритмічна структура та процедура пошуку дублікатів може бути модифікована за рахунок попереднього аналізу характеру помилки, що дозволить скоротити час на визначення рангів цільових вершин.

5. Висновок

Таким чином, в ході дослідження встановлено, що в продукційних базах знань ЗОІС можлива наявність наступних класів помилок, які виникають на етапі формування бази знань експертами:

- внутрішня суперечливість, надмірність, неповнота, репліка;
- внутрішні помилки правил бази знань;
- помилки, пов'язані із зовнішньою суперечливістю предметної області.

Для класу “Репліка” сформовані висновки щодо їх впливу на якість програмного забезпечення ЗОІС в цілому. На підставі дослідження природи зазначеного класу помилок розроблено механізм приведення логічної структури продукційної бази знань до графового вигляду. Показано, що даний клас помилки є структурним та може бути виявленим за допомогою етапу статичного зневадження на основі графових методів. Розроблено алгоритмічні структури пошуку помилок класу “Репліка” та визначена їх обчислювальна складність, зазначені алгоритми реалізовані у вигляді програмних процедур та можуть бути включені до складу інформаційної компоненти підтримки процесу розробки ЗОІС. Проведені дослідження показали, що зневадження помилок класу “Репліка” переводить продукційну базу знань до стану статичної коректності, що є необхідною умовою зневадження. Подальші напрямки досліджень можуть бути спрямовані на розробку процедур зневадження класів помилок “надмірність” та “неповнота”.

Список використаної літератури

1. Burkov V. N., Goubko M., Korgin N., Novikov D. Introduction to Theory of Control in Organizations. Boca Raton: CRC Press, 2015. 346 p.

2. Липаев В.В. Надежность и функциональная безопасность комплексов программ реального времени. М., 2013. С. 207.

3. Нариньяни А., Яхно Т. Продукционные системы. Представление знаний в человеко-машинных и робототехнических системах. М.: ВЦ АН СССР, ВИНИТИ, 1984. С. 136–177.

4. Zadeh L.A. The concept of a linguistic variable and its application to approximate reasoning. Information Sciences. 1975.Vol. 8. P. 199–249, 301–357; Vol. 9. P. 43–80.

5. Tsukamoto Y. An approach to fuzzy reasoning method. In: Gupta M.M., Ragade R.K. and Yager R.R. (eds.) Advances in fuzzy set theory and applications, 1979. P. 137–149.

6. Miller S.P., Whalen M.W., Cofer D.D. Software model checking takes off. Commun. ACM. 2010. № 53(2). P. 58–64.

7. Мейнарович Є., Кратко М. Англійсько-український словник: Математика та кібернетика. К.: Перун, 2010. 560 с.

8. Кривуля Г. Ф., Шкиль А. С., Кучеренко Д. Е. Анализ корректности продукционных правил в системах нечеткого логического вывода с использованием квантовых моделей. АСУ и приборы автоматики : всеукр. межвед. науч.-техн. сб. Х. : Изд-во ХНУРЭ, 2013. Вып. 165. С. 42–53.

9. Долинина О.Н. Информационные технологии в управлении современной организацией. 2006. Саратов: Сарат. гос. техн. ун-т. 160 с.

10. KES General Description Manual. Software Architecture and Engineering Inc. Arlington. 1983. P. 33.

11. Suwa H., Scott A.C., Shortliffe A. An Approach to Veryfing Consistency and Completeness in a Rule-Based Expert System. Rule-Based Expert Systems. London: Addison–Wesley. 1984. P. 159–170.

12. Nguen T., Perkins W., Laffey T., Pecora W. Checking Expert System Knowledge Bases for consistency and completeness. Proc. of the 9th Int. Joint Conf. on AI, Los.Ang. 1985. P. 375–378.

13. Cragun B.J., Stendel H.J. A decision-table-based processor for checking completeness and consistency in rule-based expert systems. Int. J. Man- Mach. Stud. 1987. №. 5. P. 633-648.

14. Ferrante O., Benvenuti L., Mangeruca L., Sofronis C., Ferrari A. Parallel NuSMV: A NuSMV Extension for the Verification of Complex Embedded Systems; eds. Ortmeier F., Daniel P. SAFECOMPWorkshops. 2012. Vol. 7613. P. 409-416.

15. Knauf R., Gonzalez A.J., Abel T. A framework for validation of rule-based systems. IEEE Trans Syst Man Cybern B Cybern. 2002. № 32(3). P. 281–295.

16. Godel K. Über formal unentscheidbare Sätze der Principia Mathematica und verwandter Systeme, I. and On formally undecidable propositions of Principia Mathematica and related systems I in Solomon Feferman. Kurt Gödel Collected works. Oxford University Press. 1986. P. 144–195.

17. Войтович С.А., Литвиненко М.І., Павленко М.А. Формалізований опис знань про процес відбору джерел вогневих засобів Повітряних Сил. Системи обробки інформації. 2009. № 2(76). С. 30–35.

18. Павленко М.А., Медведєв В.К., Берднік П.Г., Сафронов Р.В. Метод визначення напрямків удару засобів повітряного нападу на оперативному напрямку. Наука і техніка Повітряних Сил Збройних Сил України. 2016. № 3(24). С. 24–27.

19. Sedgewick Robert. Algorithms in C++. Parts 1-4. Fundamentals, Data Structure, Sorting, Searching. Addison-Weasley, 1998. 752 p. ISBN-10 0-201-35088-2

References

1. Burkov V. N., Goubko M., Korgin N. and Novikov D. (2015), Introduction to Theory of Control in Organizations. Boca Raton: CRC Press. 346 p.

2. Lypaev V. (2013), "Reliability and functional safety of real-time software complexes". Moscow. P. 207.

3. Naryniany A. and Yakhno T. (1984), "Production systems. Representation of knowledge in human-machine and robotic systems". Moscow: VC AN USSR, VYNYTY. P. 136–177.

4. Zadeh L.A. (1975), "The concept of a linguistic variable and its application to approximate reasoning". Information Sciences. Vol. 8. P. 199–249, 301–357; Vol. 9. P. 43–80.

5. Tsukamoto Y. (1979), "An approach to fuzzy reasoning method". In: Gupta M.M., Ragade R.K. and Yager R.R. (eds.) Advances in fuzzy set theory and applications. P. 137–149.

6. Miller S.P., Whalen M.W. and Cofer D.D. (2010), "Software model checking takes off". Commun. ACM. № 53(2). P. 58–64.

7. Mejnarovych J. and Kratko M. (2010), English-Ukrainian dictionary: Mathematics and cybernetics. Kyiv, Perun. 560 p.

8. Kryvulia G., Shkyl A. and Kucherenko D. (2013), "Analysis of the correctness of production rules in fuzzy inference systems using quantum models". ACS and automation devices: vseukr. mezhved. nauch.-tekhn. sb. Kharkiv: Yzd-vo KhNURE, Vol. 165. P. 42–53.

9. Dolynyna O. (2006), Information technology in the management of a modern organization. Saratov, SSTU. 160 p.

10. KES General Description Manual. (1983). Software Architecture and Engineering Inc. Arlington. P. 33.

11. Suwa H., Scott A.C. and Shortliffe A. (1984), "An Approach to Veryfing Consistency and Completeness in a Rule-Based Expert System". Rule-Based Expert Systems. London: Addison–Wesley. P. 159–170.

12. Nguen T., Perkins W., Laffey T. and Pecora W. (1985), "Checking Expert System Knowledge Bases for consistency and completeness". Proc. of the 9th Int. Joint Conf. on AI, Los.Ang. P. 375–378.

13. Cragun B.J. and Stendel H.J. (1987), "A decision-table-based processor for checking completeness and consistency in rule-based expert systems". Int. J. Man- Mach. Stud. №5. P. 633–648.

14. Ferrante O., Benvenuti L., Mangeruca L., Sofronis C. and Ferrari A. (2012), "Parallel NuSMV: A NuSMV Extension for the Verification of Complex Embedded Systems"; eds. Ortmeier F., Daniel P. SAFECOMPWorkshops. Vol. 7613. P. 409–416.

15. Knauf R., Gonzalez A.J. and Abel T. (2002), "A framework for validation of rule-based systems". IEEE Trans Syst Man Cybern B Cybern. №32(3). P. 281–295.

16. Godel K. (1986), "Über formal unentscheidbare Sätze der Principia Mathematica und verwandter Systeme, I. and On formally undecidable propositions of Principia Mathematica and related systems I in Solomon Feferman". Kurt Gödel Collected works. Oxford University Press. P. 144–195.

17. Voitovych S., Lytvynenko M. and Pavlenko M. (2009), "Formalized description of knowledge about the process of selection of sources of fire means of the Air Force." Information processing systems. № 2(76). P. 30–35.

18. Pavlenko M., Medvediev V., Berdnik P. and Safronov R. (2016), "The method of determining the directions of impact of air attack means in the operational direction." Science and technology of the Air Force of Ukraine. № 3(24). P. 24–27.

19. Sedgewick Robert. (1998), "Algorithms in C++. Parts 1-4. Fundamentals, Data Structure, Sorting, Searching". Addison-Weasley, P. 752.