

**Міністерство освіти і науки, молоді та спорту України
Тернопільський національний економічний університет
Факультет комп'ютерних інформаційних технологій
Кафедра комп'ютерної інженерії**

До захисту допущено
Завідувач кафедри
комп'ютерної інженерії
к.т.н., доц. О.М.Березький

_____ 20__ р.

ДИПЛОМНА РОБОТА
освітньо-кваліфікаційного рівня "Магістр"
зі спеціальності 8.05010201 "Комп'ютерні системи та мережі"
на тему:

**ЕКСПЕРТНІ МЕТОДИ ДІАГНОСТИКИ
ФУНКЦІОНАЛЬНОГО СТАНУ КОМП'ЮТЕРНИХ
СИСТЕМ В РЕАЛЬНОМУ ЧАСІ**

Студент групи КСМзм - 51
Ільчишин Т.М.

підпис

Науковий керівник
д.е.н., професор Ріппа С.П.

підпис

Консультант з нормоконтролю
Палій І.О.

Прізвище, ініціали

Підпис

Міністерство освіти і науки, молоді та спорту України
Тернопільський національний економічний університет
Факультет комп'ютерних інформаційних технологій
Кафедра комп'ютерної інженерії

“Затверджую”
Зав. кафедри
комп'ютерної інженерії
к.т.н., доц. О.М. Березький
_____”
_____ 20__ р.

З А В Д А Н Н Я
НА ДИПЛОМНУ РОБОТУ СТУДЕНТА
Ільчишина Тараса Миколайовича

- 1. Тема дипломної роботи** “Експертні методи діагностики функціонального стану комп'ютерних систем в реальному часі” затверджена наказом університету № _____ від „_____” _____ 20__ р
- 2. Термін здачі** закінченої дипломної роботи _____
- 3. Об'єкт дослідження:** процес діагностування функціонального стану комп'ютерних систем в реальному часі.
- 4. Предмет дослідження:** методи діагностики функціонального стану комп'ютерних систем експертними системами реального часу і їх апаратна реалізація.
- 5. Перелік задач, які мають бути вирішені:**
 - проаналізувати існуючі підходи до діагностики комп'ютерних систем за допомогою експертних систем;
 - вдосконалити метод діагностики комп'ютерних систем на основі представлення знань експертів у вигляді продукційних правил;
 - вдосконалити метод аналізу бази знань в експертних системах діагностики комп'ютерних систем з метою визначення можливості постановки діагнозу будь-якому стану комп'ютерної системи;
 - дослідити ефективність використання розроблених експертних методів на конкретних прикладах діагностики персонального комп'ютера;
 - здійснити апаратний синтез нейронної мережі у вигляді програмованої логічної матриці, на основі якої можна реалізувати швидкодіючу продукційну експертну систему реального часу.
- 6. Перелік ілюстративного матеріалу:**
 - схема з трьох мультиплексорів і двох суматорів,
 - продукційний граф бази знань KB₁,
 - модифікований граф бази знань KB₁,
 - продукційний граф бази знань KB₂,
 - модифікований граф бази знань KB₂
 - схема нейронної мережі,

– схема першого нейрона вхідного шару.

7. Консультанти по роботі

Розділ	Консультант	Підпис
1		
2		
3		

КАЛЕНДАРНИЙ ПЛАН

№	Назва структурних частин ДР	Термін виконання	Примітка
1	Аналіз сучасного стану проблеми діагностики комп'ютерних систем	15.09.2011 – 5.11.2011	
2	Експертні методи діагностики функціонального стану комп'ютерних систем	6.11.2011 – 31.01.2012	
3	Використання продукційних систем і бінарних відносин в експертних системах діагностики комп'ютерної техніки	1.02.2012 – 23.04.2012	

Завдання прийняв до виконання _____
(підпис)

Керівник дипломної роботи _____
(підпис)

РЕФЕРАТ

Дипломна робота на тему “Експертні методи діагностики функціонального стану комп’ютерних систем в реальному часі” на здобуття освітньо-кваліфікаційного рівня “Магістр” зі спеціальності “Комп’ютерні системи та мережі” написана обсягом 110 сторінок і містить 37 ілюстрацій, 9 таблиць, 1 додаток та 57 джерела за переліком посилань.

Метою роботи є вдосконалення методів діагностики комп’ютерних систем, направлений на підвищення ефективності процесу діагностування за рахунок удосконалення способів представлення знань експертів в експертних системах реального часу і їх апаратної реалізації.

Методи досліджень. В дипломній роботі використовуються методи дискретної математики і теорії графів, методи представлення експертних знань і штучного інтелекту, а також загальнонаукові методи досліджень – емпіричні та теоретичні, зокрема опис, аналіз, синтез, узагальнення, пояснення, систематизація, класифікація, моделювання.

Вдосконалено метод діагностики комп’ютерних систем на основі представлення знань експертів у вигляді продукційних правил та метод аналізу бази знань в експертних системах діагностики комп’ютерних систем з метою визначення можливості постановки діагнозу будь-якому стану комп’ютерної системи.

Отримані результати дозволяють створювати експертні системи діагностики комп’ютерних систем, що дозволяють поставити діагноз функціональному стану комп’ютерної системи; аналізувати продукційні експертні системи на здатність постановки діагнозу будь-якого стану комп’ютерної системи або мережі; апаратно синтезувати задану нейронну мережу у вигляді програмованої логічної матриці, на основі якої можна реалізувати швидкодіючу продукційну експертну систему реального часу.

Ключові слова: ЕКСПЕРТНА СИСТЕМА, ДІАГНОСТИКА КОМП’ЮТЕРНИХ СИСТЕМ, БАЗА ЗНАНЬ, ПРОДУКЦІЙНІ ПРАВИЛА, НЕЙРОННА МЕРЕЖА.

ABSTRACT

Diploma work «Expert methods of diagnosis of computer systems functional state in real time» on acquiring of educationally-qualification «Master's» degree, speciality «Computer systems and networks» has general volume 110 pages and contains 37 illustrations, 9 tables, 1 addition and 57 sources of information.

The object is to improve methods for diagnosis of computer systems aimed at strengthening the diagnosis by improving the methods of knowledge experts representation in real-time expert systems and their hardware implementation.

Methods of research. In the thesis work are used methods of discrete mathematics and graph theory, methods of experts knowledge representation and artificial intelligence, and methods of general scientific research – empirical and theoretical, including description, analysis, synthesis, generalization, explanation, systematization, classification, modeling.

The method for computer systems diagnosis based on expert knowledge representation in the form of production rules and the method of knowledge base analysis in expert systems, aimed at computer systems diagnosis to determine the possible diagnosis of any computer system are improved .

The obtained results allow to create the expert system diagnostic computer systems to diagnose the functional state of the computer system to analyze the productive ability of expert systems for diagnosis of any computer system or network hardware to synthesize a given neural network as a programmable logical matrix, on which you can implement fast-acting productive expert system in real time.

Keywords: EXPERT SYSTEMS, DIAGNOSIS OF COMPUTER SYSTEMS, KNOWLEDGE BASE, PRODUCTION RULES, NEURAL NETWORKS.

ЗМІСТ

Перелік умовних скорочень.....	7
Вступ.....	8
1 Аналіз сучасного стану проблеми діагностики комп'ютерних систем.....	12
1.1 Діагностика комп'ютерних систем за допомогою експертних систем.....	12
1.2 Діагностика комп'ютерної техніки експертними методами.....	16
1.3 Задачі експертної класифікації і прийняття рішень.....	21
1.4 Методи і технології штучного інтелекту.....	25
1.5 Постановка задачі дослідження.....	29
2 Експертні методи діагностики функціонального стану комп'ютерних систем.....	32
2.1 Метод діагностики комп'ютерних систем на основі продукційних систем і бінарних відносин.....	32
2.2 Метод аналізу бази знань в експертних системах діагностики комп'ютерних систем.....	52
3 Використання продукційних систем і бінарних відносин в експертних системах діагностики комп'ютерної техніки.....	63
3.1 Дослідження ефективності використання методів діагностики комп'ютерних систем.....	63
3.2 Синтез нейронної мережі в програмовану логіку.....	90
3.3 Опис синтезованої моделі нейронної мережі.....	97
Висновки.....	102
Список використаних джерел.....	104
Додаток А – Довідка про використання.....	110

ПЕРЕЛІК УМОВНИХ СКОРОЧЕНЬ

ЕС – експертна система

ДЕС – діагностична експертна система

КТ – комп'ютерна техніка

ПДЕС – продукційна діагностична експертна система

ПО – предметна область

ГКМ – глобальна комп'ютерна мережа

ПЗ – програмне забезпечення

НМ – нейронна мережа

ВСТУП

Актуальність роботи. Діагностичні експертні системи (ДЕС) призначені для виявлення джерел несправностей за наслідками спостережень за поведінкою контрольованої системи. ДЕС виконують діагностування, використовуючи опис ситуацій, характеристики поведінки або знання про конструкцію компонент, щоб встановити ймовірні причини неправильного функціонування системи, що діагностується.

При діагностиці комп'ютерної техніки (КТ) експертами часто використовується продукційна система представлення знань, які можна сформулювати у вигляді "якщо..., то...".

Важливим недоліком продукційних ДЕС (ПДЕС) є те, що існуючі методи їх побудови не гарантують класифікації кожного досліджуваного об'єкту ПДЕС. У ПДЕС, як правило, не розглядається завдання повної класифікації об'єкту дослідження. Їх головним завданням є фіксація знань експерта у визначеній ПО. Ці знання дозволяють класифікувати деякі стани об'єкту, залишаючи інші можливі стани за межами ПДЕС, що розробляється.

Інтеграція моделей представлення знань і правил навчання нейронних мереж – важливий напрям розвитку систем штучного інтелекту. Об'єднання нейронних мереж і баз знань може виконуватися різними способами. У простих випадках нейронна мережа виконує попередню обробку інформації в системах, які засновані на знаннях. В інших випадках здійснюється вбудовування нейронних мереж в бази знань, і навпаки.

Системи, які об'єднують в собі можливості, що надаються нейронними мережами і моделями представлення знань, відносять до групи гібридних інтелектуальних систем.

У разі інтеграції нейронних мереж з продукційними правилами множина продукційних правил визначає первинну структуру нейронної мережі. Після параметричного або структурного навчання мережі виконується зворотне відновлення множини правил по нейронній мережі. Це дозволяє поліпшити базу правив за допомогою алгоритмів навчання нейронних мереж.

Тому актуальною задачею є інтеграція нейронних мереж з продукційними правилами. Апаратна реалізація нейронних мереж забезпечить максимальну швидкість при вирішенні задач діагностики комп'ютерних систем в реальному часі на основі продукційних експертних систем.

Мета і завдання дослідження. Метою дипломної роботи є вдосконалення методів діагностики комп'ютерних систем, направлений на підвищення ефективності процесу діагностування за рахунок удосконалення способів представлення знань експертів в експертних системах реального часу і їх апаратної реалізації.

Для досягнення сформульованої мети необхідно вирішити наступні завдання:

- проаналізувати існуючі підходи до діагностики комп'ютерних систем за допомогою експертних систем;
- вдосконалити метод діагностики комп'ютерних систем на основі представлення знань експертів у вигляді продукційних правил;
- вдосконалити метод аналізу бази знань в експертних системах діагностики комп'ютерних систем з метою визначення можливості постановки діагнозу будь-якому стану комп'ютерної системи;
- дослідити ефективність використання розроблених експертних методів на конкретних прикладах діагностики персонального комп'ютера;
- здійснити апаратний синтез нейронної мережі у вигляді програмованої логічної матриці, на основі якої можна реалізувати швидкодіючу продукційну експертну систему реального часу.

Об'єкт дослідження – процес діагностування функціонального стану комп'ютерних систем в реальному часі.

Предмет дослідження – методи діагностики функціонального стану комп'ютерних систем експертними системами реального часу і їх апаратна реалізація.

Методи досліджень. В дипломній роботі використовуються методи дискретної математики і теорії графів, методи представлення експертних знань і штучного інтелекту, а також загальнонаукові методи досліджень – емпіричні та

теоретичні, зокрема опис, аналіз, синтез, узагальнення, пояснення, систематизація, класифікація, моделювання.

Наукова новизна одержаних результатів. Вдосконалено метод діагностики комп'ютерних систем на основі представлення знань експертів у вигляді продукційних правил та метод аналізу бази знань в експертних системах діагностики комп'ютерних систем з метою визначення можливості постановки діагнозу будь-якому стану комп'ютерної системи.

Практичне значення отриманих результатів. Отримані результати дозволяють створювати експертні системи діагностики комп'ютерних систем, що дозволяють поставити діагноз функціональному стану комп'ютерної системи; аналізувати продукційні експертні системи на здатність постановки діагнозу будь-якого стану комп'ютерної системи; апаратно синтезувати задану нейронну мережу у вигляді програмованої логічної матриці, на основі якої можна реалізувати швидкодіючу продукційну експертну систему реального часу.

В першому розділі дипломної роботи проаналізовані існуючі підходи до діагностики комп'ютерних систем за допомогою експертних систем. Цей аналіз показує, що найбільш поширеною формою представлення знань експерта є продукційні системи у поєднанні з іншими формами представлення знань. Проаналізовані сучасні методи експертної класифікації і прийняття рішень, які гарантують постановку діагнозу будь-якому стану об'єкту дослідження. Проаналізовані сучасні методи і технології штучного інтелекту. Цей аналіз показує, що важливим напрямом розвитку систем штучного інтелекту є інтеграція нейронних мереж з продукційними правилами. Тому актуальною є апаратна реалізація нейронних мереж, яка забезпечить максимальну швидкодію при вирішенні задач діагностики комп'ютерних систем в реальному часі на основі продукційних експертних систем, що гарантують постановку діагнозу будь-якому стану об'єкту дослідження.

В другому розділі роботи розглянуті продукційні експертні системи, які здатні поставити діагноз будь-якому стану комп'ютерної системи, що характеризуються одним, двома, трьома і довільним числом параметрів. Показано, що використання бінарних відносин характерності значень параметрів

для діагнозів дозволяє істотно скоротити число продукцій, необхідних для діагностики довільного стану комп'ютерної системи. Запропонований процес постановки діагнозу, який заснований на продукційному висновку і використанні бінарних відносин характерності значень параметрів для діагнозів.

В третьому розділі показана ефективність використання розроблених методів на конкретних прикладах діагностики персонального комп'ютера. Розглянутий процес апаратного синтезу заданої нейронної мережі у вигляді програмованої логічної матриці. Описана архітектура нейронної мережі, що синтезується. Приведений опис синтезованої моделі.

1 АНАЛІЗ СУЧАСНОГО СТАНУ ПРОБЛЕМИ ДІАГНОСТИКИ КОМП'ЮТЕРНИХ СИСТЕМ

1.1 Діагностика комп'ютерних систем за допомогою експертних систем

Експертна система (ЕС) CRIB надає допомогу розробникам і обслуговуючому персоналу комп'ютерів при виявленні несправностей в апаратурі і програмному забезпеченні комп'ютера [1, 2]. Розробник надає системі опис ситуацій в термінах простої англо-подібної мови. CRIB зіставляє цей опис з базою даних про відомі несправності. Послідовно зіставляючи всі великі групи симптомів з описом, що вводиться, CRIB виявляє блок, який підлягає ремонту або заміні. Якщо після ремонту або заміни несправність не усунена, система автоматично повертається до точки ухвалення останнього рішення і намагається знайти інше рішення. У системі CRIB знання і евристики по діагностиці несправностей апаратури і програмного забезпечення представлені у вигляді набору пар «симптом-дія», де дія направлена на усунення симптому. CRIB моделює машину, що діагностується, у вигляді простої ієрархії блоків семантичної мережі. Система написана на мові CORAL 66. Вона була розроблена спільно компаніями International Computers Limited, Research and Advanced Development Centre і Brunel University. CRIB досягла рівня дослідницького прототипу [3, 4].

ЕС DART (Diagnostic Assistance Reference Tool) надає допомогу при діагностуванні несправностей комп'ютерних систем, використовуючи інформацію про конструкцію пристрою, що діагностується. Система працює безпосередньо з інформацією про структуру і передбачувану поведінку пристрою і надає допомогу при пошуку проектних недоліків в наново створюваних пристроях. Система застосовується для простих обчислювальних схем і телепроцесорних пристроїв IBM 4331. DART використовує незалежну від конструкції пристрою процедуру виводу, яка аналогічна процесу доказу теорем, коли система намагається отримати доказ щодо причин несправностей пристроїв. Ця система реалізована на мові MRS і була розроблена в Станфордському університеті. Вона доведена до

рівня дослідницького прототипу [5].

ЕС IDT допомагає технічному персоналу знайти блоки, які слід замінити для усунення несправностей в EBM PDP 11/03. Система використовує знання про тестовані блоки, наприклад, функції його компонент і їх зв'язки один з одним, щоб вибрати і виконати діагностичні тести, а також інтерпретувати їх результати. Система заснована на правилах, використовує прямий ланцюжок міркувань, реалізована на FRANZ LISP і OPS 5. IDT була розроблена компанією DEC і доведена до рівня дослідницького прототипу [6].

ЕС TIMM/TUNER надає допомогу в настройці обчислювальних систем VAX/VMS для того, щоб зменшити проблеми, що виникають в обчислювальному середовищі, що постійно змінюється. Вона взаємодіє з адміністратором системи, задаючи йому ряд питань, щоб рекомендувати дії, такі, як підстроювання параметрів системи або значень, вибраних користувачем, перерозподіл або зниження вимог користувача, зміна програмою розробки користувача або придбання нової апаратури. Система використовує схему представлення знань, засновану на правилах, створену в рамках TIMM, системи комерційного рівня для автоматизації придбання знань. TIMM/TUNER була розроблена в General Research Corporation і доведена до комерційного рівня [7].

ЕС BDS надає допомогу при виявленні несправних модулів у великій підсистемі комутації сигналів, яка називається baseband distribution subsystem. Ця система використовує свідчення апаратури, щоб виявити несправні плати або інші деталі, змонтовані на шасі, із-за яких могла виникнути несправність. BDS засновує свій діагноз як на стратегіях експерта діагноста, так і на знаннях про структуру, функції і причинні зв'язки компонентів електронного пристрою. BDS реалізована на мові LES і використовує представлення знань, засноване на правилах, і управління за допомогою зворотного ланцюжка міркувань. Вона була розроблена в Lockheed Palo Alto Research Laboratory і доведена до стадії дослідницького прототипу [8].

ЕС CRITTER допомагає інженерам-розробникам інтегральних схем аналізувати коректність, тимчасові характеристики, робастність і швидкодію НВІС. У систему інженер вводить електричну схему і характеристики вхід-вихід,

по яких система буде вичерпну модель роботи схеми. CRITTER узагальнює таку інформацію і надає її інженерові з оцінкою функціональних можливостей схеми, а також інформацію по діагностуванню і ремонту. Знання, вбудовані в CRITTER, включають інформацію про принципові схеми і методи аналізу схем, таких, як моделювання підсхем і аналіз часів затримок. Принципові схеми представляються за допомогою фреймів, тоді як інші знання – за допомогою формул алгебри і числення предикатів. CRITTER реалізована на мові Інтерлісп. Вона була розроблена в Rutgers University і доведена до рівня демонстраційного прототипу [9].

ЕС DFT допомагає розробникам НВІС виявляти порушення правила DFT (design for testability, тобто в проєкті повинна бути закладена можливість тестувати схему) і переробляти проєкт так, щоб усунути ці порушення. Як тільки система виявляє порушення правила DFT, вона аналізує їх і виправляє проєкт, використовуючи знання і евристики, засновані на так званому чутливому до рівня методі сканування. DFT також витягує з проєктованої структури інформацію про управління і умови спостереження і використовує її для автоматичного породження очікуваних результатів тестування. Представлення знань в системі орієнтоване на методи логіки. Проєкт цифрової схеми складається з набору логічних тверджень, що описують з'єднання між вузлами НВІС і функції цих вузлів. DFT реалізована на мові Пролог. Вона була розроблена в Сиракузькому університеті (США) і доведена до рівня демонстраційного прототипу [10].

ЕС EL виконує аналіз стаціонарних режимів електронних схем, що складаються з резисторів, діодів і транзисторів. Отримавши опис схеми, система аналізує її і визначає значення різних її параметрів, таких як струм або напруга в заданих точках. Знання EL включають загальні закони електротехніки (наприклад, закон Ома) і характеристики елементів схеми (наприклад, що описують роботу транзистора). У системі використана схема представлення знань за допомогою правил і прямого ланцюжка міркувань, а також індексована база знань, що містить факти і твердження. EL має обмежені засоби пояснення, засновані на властивості системи запам'ятовувати обґрунтування нових тверджень. EL реалізована на мові ARS. Вона була розроблена в МТІ і доведена

до рівня дослідницького прототипу [11].

ЕС FG 502-TASP допомагає технічному персоналу виявляти неправильну роботу генераторів функцій FG 502. Система використовує емпіричні правила поведінку приладів разом з евристичними правилами, отриманими від досвідчених техніків, а не теоретичну модель функціонування FG 502, що зажадало б глибоких знань законів електроніки. З метою спрямовувати процес діагностування, система надає техніку графічне виведення розміщення деталей на потрібній друкованій платі і картину того, як повинен виглядати сигнал, поміряний у вказаній точці цієї плати. FG 502-TASP – це система, орієнтована на об'єкти, реалізована на мові SMALLTALK-80. Вона була розроблена фірмою «Tektronix» і доведена до рівня дослідницького прототипу (FG 502 function generator Troubleshooting Assistant System Prototype) [12].

ЕС FOREST локалізує і діагностує несправності в електронному устаткуванні. Система доповнює можливості виявлення і локалізації несправностей, якими володіють вживані нині діагностичні програми для автоматизованого тестуючого устаткування. Знання FOREST включають:

- емпіричні правила, запозичені у інженерів-експертів (наприклад, якщо всі вимірювання частоти проходження імпульсів не привели до успіху, слід розглянути амплітуди імпульсів);

- знання про використання принципів схем (наприклад, генерація сильного імпульсу скидає на нуль показання системних годинників);

- загальні принципи відладки електронних схем (наприклад, причиною недостатньої амплітуди сигналу може бути дуже великий опір тракту сигналу).

Ці знання кодуються у вигляді правил з коефіцієнтами впевненості, такими як використовуються в системі PROSPECTOR, і із засобами пояснення, такими як в системі тусін. Система FOREST реалізована на мові Пролог. Вона розроблена в Університеті шт. Пенсільванія в співпраці з корпорацією RCA і доведена до рівня демонстраційного прототипу [13].

ЕС PALLADIO допомагає інженерам-електронікам проектувати і тестувати нові HBIC. PALLADIO – це середовище проектування схем, яка включає:

- інтерактивні графічні редактори, що оперують електронними

компонентами високого рівня;

- редактор правил, що допомагає модифікувати функціональні характеристики елементів схеми;
- моделюючу програму, що використовує функціональні характеристики елементів схеми і опис її структури для моделювання її поведінки;
- механізми створення і уточнення параметрів проекту на різних рівнях узагальнення.

Система PALLADIO застосовувалася при проектуванні різних схем N-МОП. Вона реалізована на мові LOOPS, що забезпечує об'єктно-орієнтовані, засновані на правилах і логічні механізми уявлення. Система розроблена в Станфордському університеті і доведена до рівня дослідницького прототипу [14]. Вона названа на честь італійського архітектора Андреа Палладіо (1508-1580).

1.2 Діагностика комп'ютерної техніки експертними методами

Аналіз ЕС діагностики комп'ютерної техніки, проведений вище показує, що найбільш поширеною формою представлення знань експерта є продукційні системи.

Наприклад, універсальна мова інженерії знань OPS 5 використовує засновану на правилах схему уявлень, яка працює за допомогою прямого ланцюжка міркувань. Універсальність системи допускає різноманітні структури представлення даних і управління в рамках однієї програми. У OPS 5 є могутній механізм зіставлення із зразком і ефективний інтерпретатор для зіставлення правил з даними, але йому не вистачає розроблених засобів підтримки. У нього немає вбудованого блоку пояснення і механізмів придбання знань, і є лише мінімальні засоби редагування і відладки програм. OPS 5 є найостаннішим в серії схожих мов, заснованих на правилах (наприклад, OPS, OPS 4), які розвинулися з розробок університетом Карнегі-Меллон мов програмування, орієнтованих на моделювання людського пізнання і пам'яті.

OPS 5 і більш ранні мови з сімейства OPS були використані для багатьох програм у галузях психології пізнання, штучного інтелекту та експертних систем.

Людський досвід – це надзвичайно складна суміш теоретичних знань, евристик вирішення завдань, прикладів минулих проблем і їх рішень, навиків сприйняття і інтерпретації, а також інших аспектів, які можна описати лише як інтуїтивні. З роками людина-експерт виробляє дуже точні правила поведінки в звичайних ситуаціях. Часто ці правила приймають форму прямих асоціацій між симптомами, що спостерігаються і остаточними діагнозами, приховуючи їх глибші причинно-наслідкові зв'язки.

Наприклад [15], в експертній системі, призначеній для аналізу несправності напівпровідника на основі правил, описовий підхід дозволить будувати діагностику несправності ланцюга на таких симптомах, як знебварвлення компонентів (можливо, вказуючи на брак), історія несправностей в подібних приладах або внутрішні спостереження компоненту за допомогою електронного мікроскопа. Проте підходи, які використовують правила для зв'язку спостережень з діагнозами, є даремними для глибшого аналізу структури і функцій приладу. Точніше пояснення могло б починатися з побудови деталізованої моделі фізичної структури ланцюга і рівнянь, що описують очікувану поведінку кожного компоненту ланцюга і їх взаємодії. Діагнози могли б ґрунтуватися на сигналах, отриманих з різних точок, і моделі ланцюга для визначення точних місць несправності.

Оскільки експертні системи першого покоління ґрунтувалися на евристичних правилах, отриманих з описів методів вирішення завдань людиною-експертом, вони мали ряд обмежень [16]. Ці обмеження дозволяють подолати підходи, засновані на моделях.

Перші системи міркувань на основі моделей з'явилися в середині 1970-х і продовжували розвиватися в 1980-х роках [17]. Необхідно відзначити, що деякі з найбільш ранніх робіт були направлені на створення програмних моделей різних фізичних приладів, таких як електронні ланцюги, з метою їх вивчення [18].

Якісні міркування на основі моделей включають наступні аспекти:

- опис кожного компоненту приладу, що дозволяє моделювати їх поведінку;
- опис внутрішньої структури приладу. Воно зазвичай містить представлення компонентів і їх взаємозв'язків разом з можливістю моделювати їх

взаємодію. Необхідний ступінь знань внутрішньої структури залежить від вживаного рівня абстракції і очікуваного діагнозу;

– діагностика приватної проблеми. Це вимагає спостережень реальної поведінки приладу, зазвичай – вимірювання його входів і виходів. Вхідні і вихідні вимірювання отримати найлегше.

Завдання полягає у визначенні несправних компонентів з врахуванням спостережуваної поведінки. Це вимагає додаткових правил, що описують відомі форми несправностей для різних компонентів і їх комбінацій. Система міркувань повинна виявити найбільш ймовірні несправності, що пояснюють поведінку системи.

Щоб конкретизувати процес розробки і опису моделі, розглянемо приклад, взятий з роботи [17] – схему з трьох мультиплексорів і двох суматорів (рисунок 1.1).

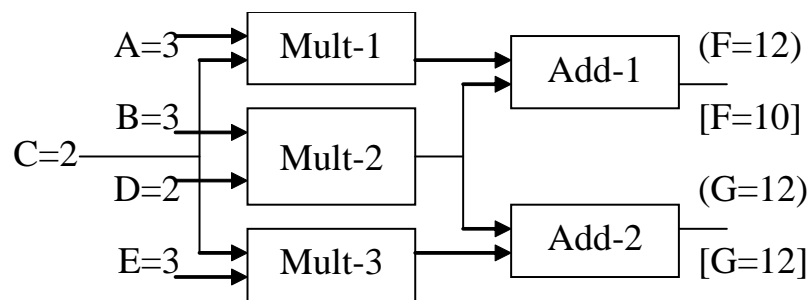


Рисунок 1.1 – Схема з трьох мультиплексорів і двох суматорів

В даному прикладі вхідні значення задаються в точках А-Е, а вихідні обчислюються в точках F і G. Очікувані вихідні значення задаються в круглих дужках, а дійсні виходи – в квадратних. Завдання полягає у визначенні несправності, що пояснює відмінність між цими значеннями.

У точці F спостерігається конфлікт: очікуваним вихідним значенням є 12, а реально отриманим – 10. Перевіряючи залежності в цій точці, визначаємо, що значення в точці F обчислюється як функція від Add-1. Це значення, у свою чергу, залежить від виходів Mult-1 і Mult-2. Один з цих пристроїв містить несправність, отже, необхідно розглянути три гіпотези: або поведінка суматора неадекватна, або некоректним є один з двох його входів. Тоді проблему треба шукати далі в

ланцюзі.

Якщо виходячи з результату F (значення 10) припустити, що поведінка Add-1 і значення його першого входу X коректні, то значення на другому вході Add-1 (позначимо його Y) повинне бути рівне 4. Але це суперечить очікуваному значенню 6, що є результатом правильної поведінки Mult-2 і входів B і D . Спостереження за цими входами показує, що на них поступали правильні значення. Отже, несправний модуль Mult-2. Варіантом другої гіпотези може бути припущення, що модуль Add-1 справний, а несправний компонент Mult-1.

Продовжуючи міркування, приходимо до висновку, що якщо на перший вхід X модуля Add-1 поступає коректне значення і сам Add-1 працює коректно, то значення другого входу Y повинне бути рівне 4. Якщо це так, то G повинно приймати значення 10, а не 12, так що вихід Mult-2 повинен дорівнювати 6 і бути коректним. Тут відкидається гіпотеза, що несправність міститься в Mult-1 або в Add-1. Ці модулі можна розглянути при подальшому тестуванні.

У міркуваннях про ситуацію, показану на рисунку 1.3, розглядалися три завдання:

- створення гіпотези про те, які компоненти пристрою могли викликати розбіжність очікуваного і реального значень;
- перевірка гіпотези, в процесі якої при заданій сукупності потенційно несправних компонентів визначається, який з них може пояснити спостережувану поведінку;
- виділення гіпотез, коли в результаті тестування залишається декілька версій і необхідно визначити, яка додаткова інформація може допомогти виявити несправність.

Нарешті, необхідно відзначити, що в розглянутому прикладі передбачається, що несправний єдиний модуль.

Дослідники продовжують заглиблювати розуміння діагностики за рахунок дослідження принципів ефективного рішення цієї задачі людиною-експертом, а також поліпшення машинної реалізації алгоритмів [19].

Евристичні правила і теоретичні моделі – два типи інформації, які людина-експерт використовує для вирішення проблем. Ще однією могутньою стратегією,

що використовується експертами, є міркування на основі досвіду вирішення завдань. У міркуваннях на основі досвіду для вирішення нових проблем використовується детальна база даних, що містить відомі вирішення завдань. Ці рішення можуть бути зібрані експертами в області знань або можуть відображати результати попередніх успішних і невдалих спроб пошуку рішення.

Області міркування на основі досвіду включають і діагностику, де часто приходять на думку невдачі минулого. Хорошим прикладом цього є також діагностика апаратних засобів. Експерт в цій області для вирішення поточної проблеми крім використання багатих теоретичних знань по електроніці і механічних системах враховує успіхи і невдачі минулого. Міркування на основі досвіду складають важливий компонент багатьох діагностичних систем, включаючи системи діагностики супутникового устаткування [20] і аналізу несправностей напівпровідникових компонентів [19].

Механізм міркування на основі досвіду функціонує таким чином:

- знаходить відповідні випадки в пам'яті;
- пристосовує знайдену ситуацію до поточної;
- застосовує перетворене рішення;
- зберігає успішне або невдале рішення для подальшого використання.

Для реалізації міркування на основі досвіду можуть бути потрібними спеціальні структури даних. Зазвичай досвід запам'ятовується як множина правил типу ситуація-дія. Факти, що описують ситуацію, є істотними ознаками цього досвіду, а оператори, що представляють дії, описують перетворення, що використовуються в новій ситуації. При використанні такого уявлення для організації і оптимізації пошуку відповідних випадків можуть застосовуватися такі алгоритми як RETE [21].

Розробка нового покоління експертних систем в області діагностики несправностей вимагає вирішення завдань, пов'язаних з адекватним інтерфейсом, із здібністю ЕС до адаптації і розпізнавання ситуацій, що знаходяться за межами даної експертизи [22]. Це пояснюється тим, що проєктовані обчислювальні системи швидко старіють, а їх складність постійно росте.

Архітектура діагностичної ЕС повинна бути заснована на структуризації і

представленні всіх типів знань, що використовуються експертом. Глобальне завдання діагностики можна розділити на деякий набір підзадач. Кожна підзадача характеризується набором умов і специфікацій, при виконанні яких рішення задачі може вважатися завершеним. Впорядкування декількох підзадач діагностики в рамках одного завдання є елементом стратегічних знань експерта.

1.3 Задачі експертної класифікації і прийняття рішень

Важливим недоліком продукційних ДЕС (ПДЕС) є те, що існуючі методи їх побудови не гарантують класифікації кожного досліджуваного об'єкту ПДЕС. У ПДЕС, як правило, не розглядається завдання повної класифікації об'єкту дослідження. Їх головним завданням є фіксація знань експерта у визначеній предметній області (ПО). Ці знання дозволяють класифікувати деякі стани об'єкту, залишаючи інші можливі стани за межами ПДЕС, що розробляється.

Очевидно, що сама постановка задачі побудови повної класифікації можлива лише тоді, коли здійснена попередня структуризація ПО (визначення множини ознак, що описують різні стани об'єкту дослідження, і бінарних відносин між ними) [23]. При цьому передбачається, що:

- різні значення атрибутів мають різні ступені характерності для несправностей, що діагностуються;
- для кожного атрибуту експерт може упорядкувати його значення по їх характерності для відповідної несправності і цей порядок не залежить від значень інших атрибутів.

Одним з таких ефективних методів побудови ДЕС є метод КЛАС [23]. Основна ідея цього методу полягає в пред'явленні експертові для класифікації деяких спеціальним чином вибраних станів об'єкту діагностування.

Істотним недоліком такого підходу є велика розмірність вектора, що описує стан складного об'єкту. На думку більшості психологів [24], процеси прийняття рішень відбуваються в короткочасній пам'яті людини. Об'єм цієї пам'яті обмежений приблизно 7-а елементами (7 ± 2). Тому класифікація станів навіть щодо простого об'єкту, опис якого містить 10 атрибутів, досвідченим експертом

може в результаті виявитися помилковою і суперечливою.

Методи ДИФКЛАСС, СТЕПКЛАСС і КЛАНШ [23], істотно ефективніші по числу звернень до експерта, чим метод КЛАС, відрізняються один від одного стратегією пред'явлення станів експертові, і теж володіють описаним вище недоліком методу КЛАС.

Завдання класифікації об'єктів, що володіють сукупністю багатьох ознак, відносяться до найбільш поширених на практиці завдань прийняття рішень. Багатокритерійні задачі класифікації відрізняються від інших багатокритерійних задач прийняття рішень тим, що в них не потрібно ранжувати альтернативи. Досить розподілити їх між декількома класами. У багатьох випадках ці класи можуть бути впорядковані за якістю, по перевазі, по ступеню вираженості деякої властивості і тому подібне.

Не дивлячись на поширеність подібного роду задач, їх теоретичне дослідження, як задач прийняття рішень, проводилося в невеликому числі робіт [24]. Одним з перших методів, призначених для вирішення задачі порядкової багатокритерійної класифікації як задачі прийняття рішення, був метод ОРКЛАСС (ОРДИНАЛЬНЯ КЛАСИФІКАЦІЯ) [25], реалізований у вигляді інтерактивної комп'ютерної системи.

Найважливішим критерієм якості алгоритму класифікації є кількість питань, ЛПР, що задаються, для отримання вирішальних правил класифікації. Час досвідченого експерта дуже цінний, тому його необхідно використовувати максимально ефективно. Одна з найбільш продуктивних ідей у напрямі мінімізації загального числа питань до ЛПР в процесі побудови повної класифікації полягає у використанні так званих ланцюгових покриттів множини векторних оцінок.

Метод класифікації, заснований на ланцюговому покритті, полягає в послідовному використанні принципу дихотомії для всіх ланцюгів покриття. Існує багато різних способів побудови ланцюгових покриттів множини векторних оцінок. Зокрема, Ж. Ансель [26], займаючись вивченням монотонних функцій алгебри логіки, довів існування і запропонував алгоритм побудови мінімального ланцюгового покриття простору, утвореного двійковими шкалами критеріїв.

В.Б. Алексеев [27] узагальнив алгоритм побудови мінімального ланцюгового покриття на випадок довільних шкал критеріїв.

В роботах [27-29] запропоновано алгоритми розшифровки монотонних функцій алгебри логіки, засновані на ланцюговому покритті простору векторних оцінок.

Важливо відзначити, що ці алгоритми використовують статичне ланцюгове покриття, тобто покриття простору будується до початку класифікації і не змінюється в процесі опитування. Однак можна припустити, що використання інформації, отриманої з відповідей ЛПР, для зміни ланцюгового покриття, дозволить сконструювати більш ефективний алгоритм, який задає менше число питань. Вперше ця ідея динамічної побудови ланцюгового покриття була застосована в алгоритмі ДІФКЛАСС [30], призначеному для побудови класифікації в задачах з двійковими шкалами критеріїв і двома класами рішень. Пізніше був розроблений алгоритм ЦКЛ (Ланцюгова Ітеративна класифікація), що дозволяє будувати повні і несуперечливі класифікації в задачах з довільними кількостями оцінок за критеріями і довільною кількістю класів рішень [31].

Методи ОРКЛАСС і ЦКЛ дозволяють побудувати класифікацію повної множини об'єктів. Проте на практиці часто потрібно класифікувати не всі можливі альтернативи, а тільки їх деяку підмножину.

Очевидно, що використання методу ОРКЛАСС або ЦКЛ для класифікації всіх можливих ситуацій неефективно, оскільки вони вимагають значного числа питань до ЛПР. Існує метод КЛАРА (КЛАСИФІКАЦІЯ РЕАЛЬНИХ АЛЬТЕРНАТИВ) [32], направлений на вирішення даного типу завдань.

Також широко поширені на практиці завдання ранжирування альтернатив, що мають оцінки по багатьом критеріям. Ці завдання характеризуються наступними особливостями:

- є достатньо велика кількість альтернатив і критеріїв;
- використовуються порядкові шкали критеріїв з вербальними оцінками;
- оцінки альтернатив можуть бути отримані тільки від людей, що грають роль «вимірювальних пристроїв»;
- правило ухвалення рішення повинне бути вироблене до появи реально

порівнюваних об'єктів.

Для вирішення подібних завдань ранжирування альтернатив раніше був розроблений метод ЗАПИТ (ЗАМКНУТІ ПРОЦЕДУРИ у ОПОРНИХ СИТУАЦІЙ) – перший з методів вербального аналізу рішень. Метод ЗАПИТ I був опублікований в 1978 р. [33]. Друга версія цього методу – ЗАПИТ II (ЛМ) [34] містила подальший розвиток ідей роботи [33]. Обидві версії були засновані на схожих процедурах отримання інформації від ЛПР, і призначені для побудови квазіпорядку на множині альтернатив.

Метод ЗАПИТ III [35] є новим етапом розвитку запропонованого раніше підходу. На відміну від попередніх версій в методі ЗАПИТ III:

- реалізована покращена процедура побудови єдиної порядкової шкали змін якості для всіх критеріїв;
- істотно зменшена кількість незрівняних альтернатив в порівнянні з методом ЗАПИТ II;
- вперше дана оцінка «вирішуючої сили» методу.

Задача ранжирування альтернатив за якістю і вибору кращими з них є близькими в тому сенсі, що послідовне виділення кращих альтернатив із заданої множини дозволяє здійснити їх ранжирування.

В рамках підходу вербального аналізу рішень раніше був запропонований метод ПАРК (ПАРНА КОМПЕНСАЦІЯ), орієнтований на вибір кращої альтернативи з групи заданих багатокритерійних альтернатив на основі їх парного порівняння, який дозволив вирішити важливі практичні завдання [36]. Проте метод ПАРК має істотні обмеження:

- метод не призначений для вибору не більше ніж 3-5 альтернатив;
- у методі використовуються парні порівняння всіх, зокрема не існуючих варіантів рішень;
- шкали критеріїв мають тільки вербальні оцінки;
- припущення про можливі процедури отримання інформації від ЛПР не підкріплені психологічними дослідженнями.

Для подолання вказаних недоліків був запропонований людино-машинний метод ШНВВ (ШКАЛА НОРМАЛІЗОВАНИХ ВПОРЯДКОВАНИХ

ВІДМІННОСТЕЙ) для вибору кращої з групи альтернатив [37], який має наступні особливості. Перш за все, в ньому активно використовуються можливості комп'ютера, який без участі ЛПР попарно порівнює всі альтернативи по певних правилах, аналізуючи їх схожість і відмінності. При цьому комп'ютер готує найбільш ефективний процес опиту ЛПР, що створює можливості для аналізу великої групи альтернатив. По-друге, деякі кількісні критерії (наприклад, вартість) також можуть розглядатися як природна і зручна мова виразу переваг ЛПР. Метод дозволяє працювати як з якісними, так і з кількісними оцінками альтернатив по критеріях. Тим самим, метод ШНВВ розширює і доповнює можливості інших методів вербального аналізу рішень.

1.4 Методи і технології штучного інтелекту

Ранні розробки штучного інтелекту були орієнтовані на створення розумних машин, які копіювали поведінку людини, проте в даний час більшість дослідників і розробників штучного інтелекту переслідують більш практичні цілі. У число прикладних алгоритмів разом з системами, заснованими на продукційних правилах [38], входять [39]:

- нейронні мережі;
- генетичні алгоритми;
- нечітка логіка;
- розумні агенти;
- мурашкові алгоритми.

Відносно систем штучного інтелекту іноді можна почути наступні критичні зауваження:

- такі системи дуже «крихкі» в тому сенсі, що, зустрівшись з ситуацією, не передбаченою розробником, вони або формують повідомлення про помилки, або дають неправильні результати (іншими словами, ці програми досить просто можна «поставити в безвихідь»);
- вони не здатні безперервно самонавчатися, як це робить людина в процесі вирішення виникаючих проблем.

Ще в середині 1980-х років багато дослідників рекомендували використовувати для подолання цих і інших недоліків нейронні мережі.

У найпростішому вигляді нейронну мережу можна розглядати як спосіб моделювання в технічних системах принципів організації і механізмів функціонування головного мозку людини. Згідно сучасним уявленням, кора головного мозку людини є множиною взаємозв'язаних простих осередків – нейронів, кількість яких оцінюється числом порядку 10^{10} . Технічні системи, в яких робиться спроба відтворити, нехай і в обмежених масштабах, подібну структуру (апаратно або програмно), отримали назву нейронні мережі [40].

Генетичні алгоритми пропонують модель оптимізації, яку можна застосовувати при вирішенні як числових, так і символічних завдань. Генетичне програмування використовується, наприклад, при створенні послідовності інструкцій. Подібні послідовності застосовуються при вирішенні математичних завдань [10].

Генетичні алгоритми відображають принципи природного відбору і генетики: виживання найбільш перспективних особин, спадкоємство і мутації. При цьому людина не втручається в процес пошуку, але може опосередковано впливати на нього заданням певних параметрів.

Як і всі методи випадкового пошуку, генетичні алгоритми орієнтовані на знаходження не оптимального рішення, а на пошук кращого, ніж існуючі на даний момент рішення. Такий підхід ефективний для складних систем, де найчастіше необхідно якимось чином поліпшити поточне рішення, а завдання пошуку оптимального рішення не ставиться через складність системи і, як наслідок, неможливість застосування традиційних методів, які спрямовані на знаходження оптимальних рішень [41].

Ту роль, яку в класичній теорії множин відіграє двозначна булева логіка, в теорії нечітких множин відіграє багатозначна нечітка логіка, в якій припущення про належність об'єкта множині можуть приймати дійсні значення в інтервалі від 0 до 1.

Аналоги операцій кон'юнкції і диз'юнкції в нечіткій логіці не мають ніякого зв'язку з теорією ймовірностей [42]. Оператори мають властивості

комутативності, асоціативності та взаємної дистрибутивності. Як до операторів в стандартній логіці, до них застосуємо принцип композитивності, тобто значення складових виразів обчислюються тільки за значеннями виразів-компонентів. У цьому оператори нечіткої логіки становлять повну протилежність законам теорії ймовірностей, згідно з якими при обчисленні ймовірностей кон'юнкції і диз'юнкції величин потрібно брати до уваги умовні ймовірності [43].

Нечітка логіка має справу з ситуаціями, коли і сформульоване питання, і знання, якими володіють, містять нечітко окреслені поняття. Проте нечіткість формулювання понять є не єдиним джерелом невизначеності. Іноді просто немає впевненості в самих фактах. Теорія можливостей є одним з напрямків в нечіткій логіці, в якому розглядаються точно сформульовані питання, що базуються на неточних знаннях.

На основі нечіткої логіки часто будуються системи управління, що містять не тільки стандартні оператори нечіткої логіки, але і допоміжні функції, які підтримують створення функцій нечіткої логіки [45].

Агент – це апаратна або програмна суть, здатна діяти на користь досягнення цілей, поставлених перед ним власником і/або користувачем [46].

До побудови агентно-орієнтованих систем можна вказати два підходи: реалізація єдиного автономного агента або розробка мультиагентної системи. Автономний агент взаємодіє тільки з користувачем і реалізує весь спектр функціональних можливостей, необхідних в рамках агентно-орієнтованої програми. На противагу цьому мультиагентні системи є програмно-обчислювальними комплексами, де взаємодіють різні агенти для вирішення завдань, які важкі або недоступні через свою складність для одного агента. Часто такі мультиагентні системи називають агентствами, в рамках яких агенти спілкуються, кооперуються і домовляються між собою для пошуку рішення поставленої перед ними задачі [1].

Мурашкові алгоритми – це порівняльно новий метод, який може використовуватися для пошуку оптимальних шляхів по графові. Дані алгоритми симулюють рух мурашок в навколишньому середовищі і використовують модель ферментів для комунікації з іншими агентами [46].

Мурашкові алгоритми цікаві, оскільки відображають ряд специфічних властивостей, властивих самим мурашкам. Мурашки легко вступають в співпрацю і працюють разом для досягнення загальної мети. Мурашкові алгоритми працюють так само, як мурашки. Це виражається в тому, що змодельовані мурашки спільно вирішують проблему і допомагають іншим мурашкам в подальшій оптимізації рішення. У цьому і полягає базова ідея мурашкових алгоритмів – оптимізація шляхом непрямого зв'язку між автономними агентами.

Інтеграція моделей представлення знань і правил навчання нейронних мереж – важливий напрям розвитку систем штучного інтелекту. Об'єднання нейронних мереж і баз знань може виконуватися різними способами. У простих випадках нейронна мережа виконує попередню або завершуючу обробку інформації в системах, заснованих на знаннях. У цікавіших випадках здійснюється вбудовування нейронних мереж в бази знань, і навпаки.

Системи, об'єднуючі в собі можливості, що надаються нейронними мережами і моделями представлення знань, відносять до групи гібридних інтелектуальних систем.

У разі інтеграції нейронних мереж з продукційними правилами множина продукційних правил визначає первинну структуру нейронної мережі. Після параметричного або структурного навчання мережі виконується зворотне відновлення множини правил по нейронній мережі. Це дозволяє поліпшити базу правил за допомогою алгоритмів навчання нейронних мереж. Для реалізації такого підходу запропоновано використовувати правила у формі диз'юнктивів Хорна. Множина правил утворює мережу. Після навчання структура нейронної мережі представляється у вигляді множини правил. Для такого перетворення використовується КТ-алгоритм.

1.5 Постановка задачі дослідження

Діагностичні ЕС (ДЕС) призначені для виявлення джерел несправностей за наслідками спостережень за поведінкою контрольованої системи. ДЕС виконують діагностування, використовуючи опис ситуацій, характеристики поведінки або знання про конструкцію компонент, щоб встановити ймовірні причини неправильного функціонування системи, що діагностується.

Наприклад, аналізатори глобальної комп'ютерної мережі (ГКМ) перевіряють і аналізують угоди про рівень сервісу, статистику і проблеми за тривалий період, а також випадки перевантаження каналів. Для оцінки і пошуку помилок аналізатор ГКС, що складається, як правило, з апаратної і програмної частин, повинен серед іншого мати функцію ДЕС реального часу. База знань для діагностики в реальному часі з пропозицією можливих рішень спрощує пошук помилок і тим самим підвищує готовність мережі. Відображення експертних повідомлень має сенс для представлення критичних аномалій глобальної мережі [47].

При діагностиці комп'ютерної техніки (КТ) експертами часто використовується продукційна система представлення знань, які можна сформулювати у вигляді "якщо..., то...". Наприклад:

– "якщо об'єкт має несправні елементи і для спостережуваного виходу виконується умова..., тоді множина... містить хоч би один несправний елемент" [48];

– "якщо після повторного виконання сеансу аналізу протоколів помилка на інший пристрій не перейшла..., перевірте відповідну абонентську область..." [49].

Важливим недоліком продукційних ДЕС (ПДЕС) є те, що існуючі методи їх побудови не гарантують класифікації кожного досліджуваного об'єкту ПДЕС. У ПДЕС, як правило, не розглядається завдання повної класифікації об'єкту дослідження. Їх головним завданням є фіксація знань експерта у визначеній ПО. Ці знання дозволяють класифікувати деякі стани об'єкту, залишаючи інші можливі стани за межами ПДЕС, що розробляється.

Метою дипломної роботи є вдосконалення методів діагностики

комп'ютерних систем, направлений на підвищення ефективності процесу діагностування за рахунок удосконалення способів представлення знань експертів в експертних системах реального часу і їх апаратної реалізації.

Для досягнення сформульованої мети необхідно вирішити наступні завдання:

- проаналізувати існуючі підходи до діагностики комп'ютерних систем за допомогою експертних систем;
- розробити метод діагностики комп'ютерних систем на основі представлення знань експертів у вигляді продукційних правил;
- розробити метод аналізу бази знань в експертних системах діагностики комп'ютерних систем з метою визначення можливості постановки діагнозу будь-якому стану комп'ютерної системи;
- дослідити ефективність використання розроблених експертних методів на конкретних прикладах діагностики персонального комп'ютера;
- здійснити апаратний синтез заданої нейронної мережі у вигляді програмованої логічної матриці, на основі якої можна реалізувати швидкодіючу продукційну експертну систему реального часу.

Отже, в першому розділі проаналізовані існуючі підходи до діагностики комп'ютерних систем за допомогою експертних систем. Цей аналіз показує, що найбільш поширеною формою представлення знань експерта є продукційні системи у поєднанні з іншими формами представлення знань (моделі, міркування на основі досвіду, фрейми і т.д.).

Важливим недоліком продукційних діагностичних експертних систем є те, що існуючі методи їх побудови не гарантують діагностику довільного стану досліджуваного ними об'єкту. Для діагностики комп'ютерних систем в реальному часі це досконало неприпустимо. Проаналізовані сучасні методи експертної класифікації і прийняття рішень, які гарантують постановку діагнозу будь-якому стану об'єкту дослідження. Цей аналіз показує, що при великій розмірності вектора, що описує стан таких складних об'єктів як комп'ютерна система або мережа, ці методи складні навіть для досвідченого експерта і результати можуть бути помилкові і суперечливі. Тому актуальним є використання продукційних

систем (як найбільш прийнятної форми представлення знань експерта в процесі діагностики комп'ютерних систем і мереж) у поєднанні з методами експертної класифікації, що гарантують постановку діагнозу будь-якому стану об'єкту дослідження.

Проаналізовані сучасні методи і технології штучного інтелекту. Цей аналіз показує, що важливим напрямом розвитку систем штучного інтелекту є інтеграція нейронних мереж з продукційними правилами. Тому актуальною є апаратна реалізація нейронних мереж, яка забезпечить максимальну швидкість при вирішенні задач діагностики комп'ютерних систем в реальному часі на основі продукційних експертних систем, що гарантують постановку діагнозу будь-якому стану об'єкту дослідження.

2 ЕКСПЕРТНІ МЕТОДИ ДІАГНОСТИКИ ФУНКЦІОНАЛЬНОГО СТАНУ КОМП'ЮТЕРНИХ СИСТЕМ

2.1 Метод діагностики комп'ютерних систем на основі продукційних систем і бінарних відносин

Діагностика комп'ютерної техніки з одним параметром. Нехай комп'ютерна система S характеризується тільки одним параметром a , який може приймати всього три значення: a_1 , a_2 і a_3 . Нехай множина значень діагнозу d , який може бути поставлений системі S , складається з двох елементів:

- d_0 – система S справна;
- d_1 – система S несправна.

База знань KB_1 продукційної експертної системи реального часу, здатна поставити діагноз будь-якому стану системи S , може складатися, наприклад, з наступної множини P продукцій:

- p_1 – «Якщо $a = a_1$, то $d = d_0$ »;
- p_2 – «Якщо $a = a_2$, то $d = d_1$ »;
- p_3 – «Якщо $a = a_3$, то $d = d_1$ ».

Продукційний граф, що описує дану базу знань, представлений на рисунку 2.1.

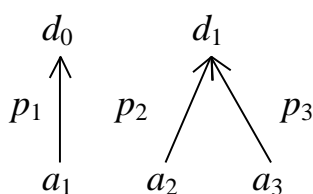


Рисунок 2.1 – Продукційний граф бази знань KB_1

Експертові, знання якого зафіксовані в базі знань KB_1 , можна поставити наступне питання: «Яке значення параметра a є найхарактернішим для діагнозу d_1 ?»

Нехай експерт відповів, що значення a_3 параметра a найхарактерніший для діагнозу d_1 , чим значення a_2 . Тоді це бінарне відношення r_1 характерності значень

параметра a для діагнозу d_1 можна представити графічно так, як показано на рисунку 2.2.

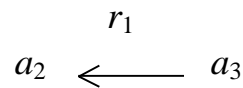


Рисунок 2.2 – Відношення характерності r_1

На рисунку 2.2 стрілка направлена від характернішого значення a_3 параметра a для діагнозу d_1 до менш характерного значення a_2 .

Ця відповідь експерта показує, що продукція p_2 представляє менш очевидне, але тому цінніше знання експерта, яке він використовує при діагностиці комп'ютерної системи S . Значення a_2 параметра a є в деякому розумінні «прикордонним»: його зміна може перевести систему S із стану, що характеризується діагнозом d_1 , в стан, який характеризується діагнозом d_2 . Останнє твердження виглядатиме очевиднішим, якщо у відношення r_1 характерності ввести і значення a_1 . Таке доповнене відношення показане на рисунку 2.3.

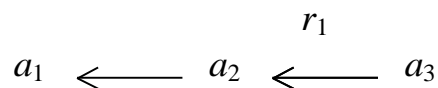


Рисунок 2.3 – Доповнене відношення характерності r_1

Таким чином, з бази знань KB_1 можна виключити продукцію p_3 , замінивши її відношенням r_1 .

Граф, що описує дану базу знань, представлену продукціями і бінарним відношенням, показаний на рисунку 2.4.

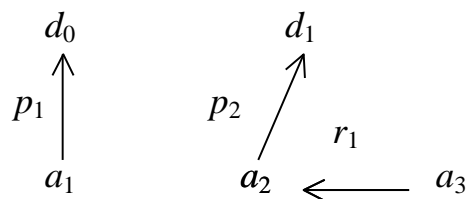


Рисунок 2.4 – Модифікований граф бази знань KB_1

Постановка діагнозу в експертній системі, заснованій на такій базі знань,

відбувається по продукційних правилах з врахуванням відношення характерності. Наприклад, якщо на вхід ЕС поступає значення a_3 , то, використовуючи відношення r_1 , можна перейти до значення a_2 , на основі якого можна зробити висновок, використовуючи стандартні засоби продукційних моделей.

Нехай знову комп'ютерна система S характеризується тільки одним параметром a , але який тепер може приймати вже дев'ять значень: $a_1, a_2, a_3, a_4, a_5, a_6, a_7, a_8$ і a_9 . Нехай множина значень діагнозу d , який може бути поставлений системі S , складається з трьох елементів:

- d_0 – система S справна;
- d_1 – у системи S несправність одного типу;
- d_2 – у системи S несправність іншого типу.

База знань KB_2 продукційної експертної системи реального часу, здатна поставити діагноз будь-якому стану системи S , може складатися, наприклад, з наступної множини P продукцій:

- p_1 – «Якщо $a = a_1$, то $d = d_0$ »;
- p_2 – «Якщо $a = a_2$, то $d = d_0$ »;
- p_3 – «Якщо $a = a_3$, то $d = d_0$ »;
- p_4 – «Якщо $a = a_4$, то $d = d_1$ »;
- p_5 – «Якщо $a = a_5$, то $d = d_1$ »;
- p_6 – «Якщо $a = a_6$, то $d = d_1$ »;
- p_7 – «Якщо $a = a_7$, то $d = d_2$ »;
- p_8 – «Якщо $a = a_8$, то $d = d_2$ »;
- p_9 – «Якщо $a = a_9$, то $d = d_2$ ».

Продукційний граф, що описує дану базу знань, представлений на рисунку 2.5.

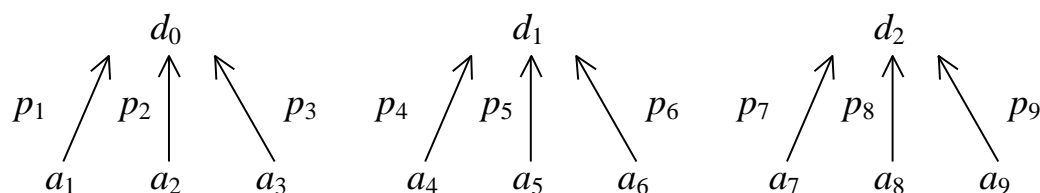


Рисунок 2.5 – Продукційний граф бази знань KB_2

Експертові, знання якого зафіксовані в базі знань KB_2 , можна поставити наступні два питання:

- яке значення параметра a характерніше для діагнозу d_1 ;
- яке значення параметра a характерніше для діагнозу d_2 .

Нехай експерт відповів, що значення a_6 параметра a характерніше для діагнозу d_1 , ніж значення a_5 , яке, у свою чергу, характерніше для діагнозу d_1 , ніж значення a_4 .

Нехай експерт відповів, що значення a_9 параметра a характерніше для діагнозу d_2 , ніж значення a_8 , яке, у свою чергу, характерніше для діагнозу d_2 , ніж значення a_7 .

Ці бінарні відносини r_1 і r_2 характерності значень параметра a для діагнозів d_1 і d_2 відповідно можна представити графічно так, як показано на рисунку 2.6.

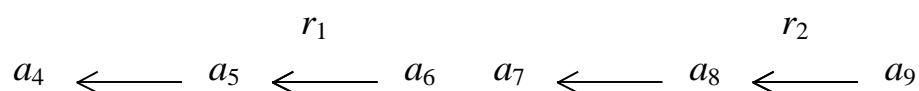


Рисунок 2.6 – Відносини характерності r_1 і r_2

На рисунку 2.6 стрілки направлені від характерніших значень параметра a для діагнозів d_1 і d_2 до менш характерних значень.

Ці відповіді експерта показують, що продукції p_4 і p_7 представляють менш очевидні, але тому цінніші знання експерта, які він використовує при діагностиці комп'ютерної системи S . Значення a_4 і a_7 параметра a є в деякому розумінні «прикордонними»: їх зміни можуть перевести систему S із стану, що характеризується одним діагнозом, в стан, який характеризується іншим діагнозом. Останнє твердження виглядатиме очевиднішим, якщо у відносини r_1 і r_2 характерності ввести і інші значення параметра a . Такі доповнені відносини показані на рисунку 2.7.

Таким чином, з бази знань KB_2 можна виключити продукції p_5 і p_6 , а також p_8 і p_9 , замінивши їх відносинами r_1 і r_2 відповідно.

Відмітимо також, що експертові, знання якого зафіксовані в базі знань KB_2 , можна також поставити наступне питання: «Яке значення параметра a характерніше для діагнозу d_0 ?»

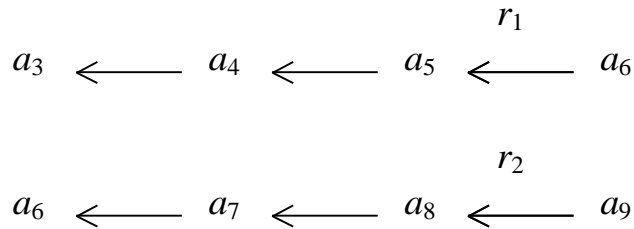


Рисунок 2.7 – Доповнені відносини характерності r_1 і r_2

Нехай експерт-діагност відповів, що значення a_3 параметра a характерніше для діагнозу d_0 , ніж значення a_2 , яке, у свою чергу, характерніше для діагнозу d_0 , ніж значення a_1 . Тоді це бінарне відношення r_0 характерності значень параметра a для діагнозу d_0 можна представити графічно так, як показано на рисунку 2.8.

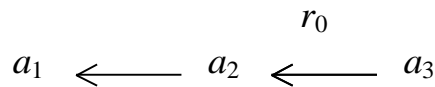


Рисунок 2.8 – Відношення характерності r_0

На рисунку 2.8 стрілки направлені від характерніших значень параметра a для діагнозу d_0 до менш характерних значень.

Ця відповідь експерта показує, що продукція p_1 представляє менш очевидне, але тому цінніше знання експерта, яке він використовує при діагностиці комп'ютерної системи S . Значення a_1 параметра a є в деякому розумінні «прикордонним»: його зміна може перевести систему S із справного стану в один з двох несправних станів. Останнє твердження виглядатиме очевиднішим, якщо у відношення r_0 характерності ввести і інші значення параметра a . Таке доповнене відношення показане на рисунку 2.9.

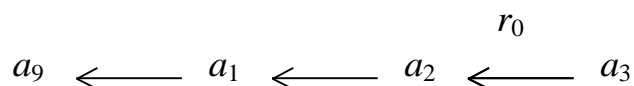


Рисунок 2.9 – Доповнене відношення характерності r_0

Таким чином, з бази знань KB_2 можна виключити продукції p_2 і p_3 , замінивши їх відношенням r_0 .

Граф, що описує дану базу знань, представлену продукціями і бінарними

відносинами r_0 , r_1 і r_2 , показаний на рисунку 2.10.

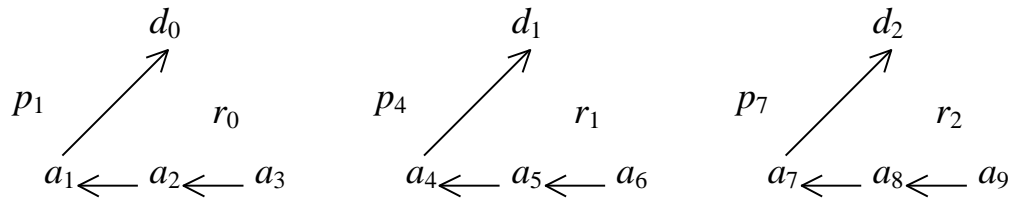


Рисунок 2.10 – Модифікований граф бази знань KB_2

Постановка діагнозу в ЕС, яка заснована на базі знань KB_2 , відбувається, наприклад, таким чином: якщо на вхід ЕС поступає значення a_5 , то, використовуючи відношення r_1 , можна перейти до значення a_4 , на основі якого можна поставити діагноз, використовуючи стандартний продукційний висновок.

Діагностика комп'ютерної техніки з двома параметрами. Нехай комп'ютерна система S характеризується двома параметрами a і b . Параметр a може приймати чотири значення: a_1, a_2, a_3, a_4 . Параметр b може приймати три значення: b_1, b_2, b_3 . Нехай множина значень діагнозу d , який може бути поставлений системі S , складається з трьох елементів:

- d_0 – система S справна;
- d_1 – у системи S несправність одного типу;
- d_2 – у системи S несправність іншого типу.

База знань KB_3 продукційної експертної системи реального часу, здатна поставити діагноз будь-якому стану системи S , може полягати, наприклад, з наступної множини P продукцій:

- p_1 – «Якщо $a = a_1$ і $b = b_1$, то $d = d_1$ »;
- p_2 – «Якщо $a = a_1$ і $b = b_2$, то $d = d_1$ »;
- p_3 – «Якщо $a = a_1$ і $b = b_3$, то $d = d_0$ »;
- p_4 – «Якщо $a = a_2$ і $b = b_1$, то $d = d_1$ »;
- p_5 – «Якщо $a = a_2$ і $b = b_2$, то $d = d_1$ і d_2 »;
- p_6 – «Якщо $a = a_2$ і $b = b_3$, то $d = d_2$ »;
- p_7 – «Якщо $a = a_3$ і $b = b_1$, то $d = d_0$ »;
- p_8 – «Якщо $a = a_3$ і $b = b_2$, то $d = d_2$ »;
- p_9 – «Якщо $a = a_3$ і $b = b_3$, то $d = d_2$ »;

- p_{10} – «Якщо $a = a_4$ і $b = b_1$, то $d = d_0$ »;
- p_{11} – «Якщо $a = a_4$ і $b = b_2$, то $d = d_2$ »;
- p_{12} – «Якщо $a = a_4$ і $b = b_3$, то $d = d_2$ ».

Продукційний граф, що описує дану базу знань, представлений на рисунку 2.11.

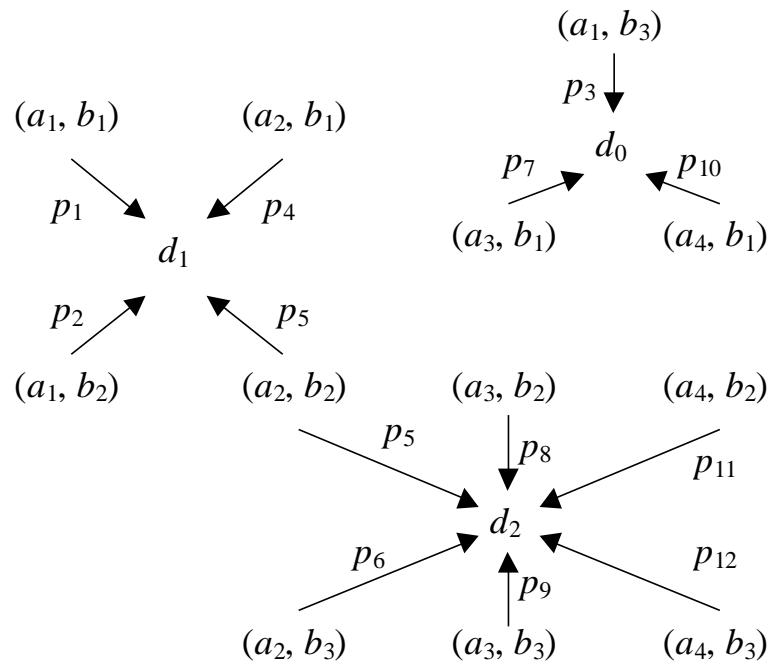


Рисунок 2.11 – Продукційний граф бази знань KB_3

Експертові, знання якого зафіксовані в базі знань KB_3 , можна поставити наступні чотири питання:

- яке значення параметра a характерніше для діагнозу d_1 ;
- яке значення параметра b характерніше для діагнозу d_1 ;
- яке значення параметра a характерніше для діагнозу d_2 ;
- яке значення параметра b характерніше для діагнозу d_2 .

Нехай експерт відповів, що значення a_1 параметра a характерніше для діагнозу d_1 , ніж значення a_2 . Нехай експерт відповів, що значення b_1 параметра b характерніше для діагнозу d_1 , ніж значення b_2 .

Нехай експерт відповів, що значення a_4 параметра a характерніше для діагнозу d_2 , ніж значення a_3 , яке, у свою чергу, характерніше для діагнозу d_2 , ніж значення a_2 . Нехай експерт відповів, що значення b_3 параметра b характерніше для діагнозу d_2 , ніж значення b_2 .

Ці бінарні відносини r_{a1} і r_{a2} характерності значень параметра a для діагнозів d_1 і d_2 відповідно, а також бінарні відносини r_{b1} і r_{b2} характерності значень параметра b для діагнозів d_1 і d_2 відповідно можна представити графічно так, як показано на рисунку 2.12.

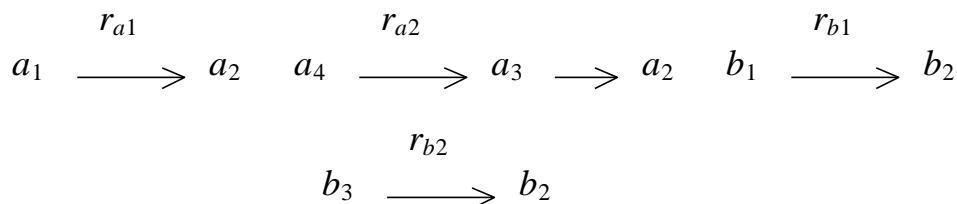


Рисунок 2.12 – Відносини характерності r_{a1} , r_{a2} , r_{b1} і r_{b2}

На рисунку 2.12 стрілки направлені від характерніших значень параметрів a і b для діагнозів d_1 і d_2 до менш характерних значень.

Ці відповіді експерта показують, що продукція p_5 представляє менш очевидне, але тому цінніше знання експерта, яке він використовує при діагностиці комп'ютерної системи S . Значення a_2 параметра a і значення b_2 параметра b є в деякому розумінні «прикордонними»: їх зміни можуть перевести систему S із стану, що характеризується одним діагнозом, в стан, який характеризується іншим діагнозом.

Таким чином, з бази знань KB_3 можна виключити продукції p_1 , p_2 і p_4 , а також p_6 , p_8 , p_9 , p_{11} і p_{12} , замінивши їх відносинами r_{a1} , r_{b1} і r_{a2} , r_{b2} відповідно. Граф, що описує дану базу знань, представлений продукціями і бінарними відносинами r_{a1} , r_{b1} і r_{a2} , r_{b2} , показаний на рисунку 2.13.

Постановка діагнозу в ЕС, яка заснована на базі знань KB_3 , відбувається, наприклад, таким чином: якщо на вхід ЕС поступають значення a_4 і b_3 та використовуючи відносини r_{a2} і r_{b2} , можна перейти до значень a_2 і b_2 , на основі яких можна ставити діагноз, використовуючи стандартний продукційний висновок.

Нехай комп'ютерна система S характеризується двома параметрами a і b . Параметр a може приймати п'ять значень: a_1 , a_2 , a_3 , a_4 , a_5 . Параметр b може приймати п'ять значень: b_1 , b_2 , b_3 , b_4 , b_5 . Нехай множина значень діагнозу d , який може бути поставлений системі S , складається з шести елементів:

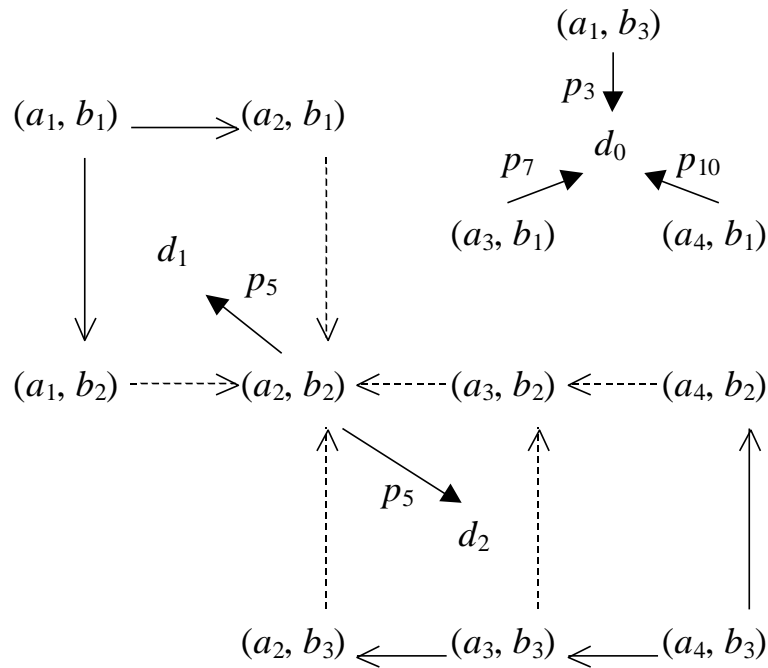


Рисунок 2.13 – Модифікований граф бази знань KB_3

- d_0 – система S справна;
- d_1 – у системи S несправність першого типу;
- d_2 – у системи S несправність другого типу;
- d_3 – у системи S несправність третього типу;
- d_4 – у системи S несправність четвертого типу;
- d_5 – у системи S несправність п'ятого типу.

База знань KB_4 продукційної експертної системи реального часу, здатна поставити діагноз будь-якому стану системи S , може полягати, наприклад, з наступної множини P продукцій:

- p_1 – «Якщо $a = a_1$ і $b = b_1$, то $d = d_1$ і d_2 »;
- p_2 – «Якщо $a = a_1$ і $b = b_2$, то $d = d_1, d_2, d_3$ і d_5 »;
- p_3 – «Якщо $a = a_1$ і $b = b_3$, то $d = d_2, d_3$ і d_4 »;
- p_4 – «Якщо $a = a_1$ і $b = b_4$, то $d = d_1, d_2$ і d_3 »;
- p_5 – «Якщо $a = a_1$ і $b = b_5$, то $d = d_1, d_2, d_3, d_4$ і d_5 »;
- p_6 – «Якщо $a = a_2$ і $b = b_1$, то $d = d_1$ »;
- p_7 – «Якщо $a = a_2$ і $b = b_2$, то $d = d_1, d_3$ і d_5 »;
- p_8 – «Якщо $a = a_2$ і $b = b_3$, то $d = d_2, d_3, d_4$ і d_5 »;
- p_9 – «Якщо $a = a_2$ і $b = b_4$, то $d = d_1$ і d_3 »;

- p_{10} – «Якщо $a = a_2$ і $b = b_5$, то $d = d_1, d_3, d_4$ і d_5 »;
- p_{11} – «Якщо $a = a_3$ і $b = b_1$, то $d = d_1, d_2$ і d_4 »;
- p_{12} – «Якщо $a = a_3$ і $b = b_2$, то $d = d_2$ і d_4 »;
- p_{13} – «Якщо $a = a_3$ і $b = b_3$, то $d = d_2$ і d_4 »;
- p_{14} – «Якщо $a = a_3$ і $b = b_4$, то $d = d_1, d_2, d_3$ і d_4 »;
- p_{15} – «Якщо $a = a_3$ і $b = b_5$, то $d = d_4$ »;
- p_{16} – «Якщо $a = a_4$ і $b = b_1$, то $d = d_0$ »;
- p_{17} – «Якщо $a = a_4$ і $b = b_2$, то $d = d_3$ і d_5 »;
- p_{18} – «Якщо $a = a_4$ і $b = b_3$, то $d = d_2, d_3, d_4$ і d_5 »;
- p_{19} – «Якщо $a = a_4$ і $b = b_4$, то $d = d_3$ »;
- p_{20} – «Якщо $a = a_4$ і $b = b_5$, то $d = d_3, d_4$ і d_5 »;
- p_{21} – «Якщо $a = a_5$ і $b = b_1$, то $d = d_1$ і d_5 »;
- p_{22} – «Якщо $a = a_5$ і $b = b_2$, то $d = d_5$ »;
- p_{23} – «Якщо $a = a_5$ і $b = b_3$, то $d = d_2, d_4$ і d_5 »;
- p_{24} – «Якщо $a = a_5$ і $b = b_4$, то $d = d_5$ »;
- p_{25} – «Якщо $a = a_5$ і $b = b_5$, то $d = d_4$ і d_5 ».

Експертові, знання якого зафіксовані в базі знань KB_4 , можна поставити наступні десять питань:

- яке значення параметра a характерніше для діагнозу d_1 ;
- яке значення параметра b характерніше для діагнозу d_1 ;
- яке значення параметра a характерніше для діагнозу d_2 ;
- яке значення параметра b характерніше для діагнозу d_2 ;
- яке значення параметра a характерніше для діагнозу d_3 ;
- яке значення параметра b характерніше для діагнозу d_3 ;
- яке значення параметра a характерніше для діагнозу d_4 ;
- яке значення параметра b характерніше для діагнозу d_4 ;
- яке значення параметра a характерніше для діагнозу d_5 ;
- яке значення параметра b характерніше для діагнозу d_5 .

Нехай експерт відповів, що значення a_2 параметра a характерніше для діагнозу d_1 , ніж значення a_1 , яке, у свою чергу, характерніше для діагнозу d_1 , ніж значення a_3 , характерніше для діагнозу d_1 , ніж значення a_5 .

Нехай експерт відповів, що значення b_1 параметра b характерніше для діагнозу d_1 , ніж значення b_4 , яке, у свою чергу, характерніше для діагнозу d_1 , ніж значення b_2 , характерніше для діагнозу d_1 , ніж значення b_5 .

Бінарні відносини r_{a1}, r_{b1} характерності значень параметрів a, b для діагнозу d_1 можна представити графічно так, як показано на рисунку 2.14.



Рисунок 2.14 – Відносини характерності r_{a1} і r_{b1}

Відносини r_{a1}, r_{b1} дозволяють побудувати бінарне відношення r_1 на множині станів системи S , якій може бути поставлений діагноз d_1 . Відношення r_1 дозволяє ставити діагноз d_1 тільки за допомогою продукцій p_5, p_{14} і p_{21} (рисунок 2.15). Постановка діагнозу в ЕС, яка заснована на модифікованій таким чином базі знань KB_4 , відбувається, наприклад, таким чином: якщо на вхід ЕС поступає стан (a_2, b_1) то, використовуючи відношення r_1 , можна перейти до стану (a_1, b_5) , на основі якого можна ставити діагноз, використовуючи стандартний продукційний висновок.

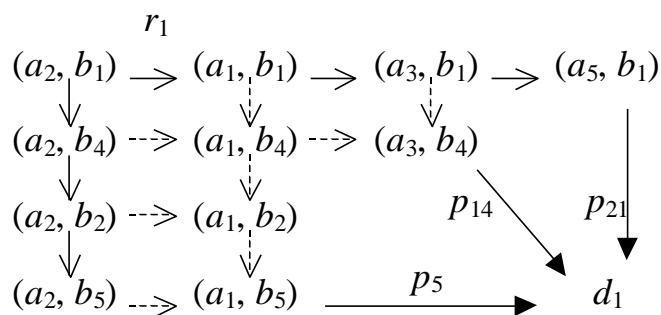


Рисунок 2.15 – Постановка діагнозу d_1

Нехай експерт відповів, що значення a_1 параметра a характерніше для діагнозу d_2 , ніж значення a_3 , яке, у свою чергу, характерніше для діагнозу d_2 , ніж значення a_5 , характерніше для діагнозу d_2 , ніж значення a_4 . Найхарактернішим значенням параметра a для діагнозу d_2 є значення a_2 .

Нехай експерт відповів, що значення b_3 параметра b характерніше для діагнозу d_2 , ніж значення b_1 , яке, у свою чергу, характерніше для діагнозу d_2 , ніж

значення b_4 , характерніше для діагнозу d_2 , ніж значення b_2 . Найхарактернішим значенням параметра b для діагнозу d_2 є значення b_5 .

Бінарні відносини r_{a2} , r_{b2} характерності значень параметрів a , b для діагнозу d_2 можна представити графічно так, як показано на рисунку 2.16.



Рисунок 2.16 – Відносини характерності r_{a2} і r_{b2}

Відносини r_{a2} , r_{b2} дозволяють побудувати бінарне відношення r_2 на множині станів системи S , якій може бути поставлений діагноз d_2 . Відношення r_2 дозволяє ставити діагноз d_2 тільки за допомогою продукцій p_5 , p_8 і p_{12} (рисунок 2.17). Постановка діагнозу в ЕС, яка заснована на модифікованій таким чином базі знань KB_4 , відбувається, наприклад, таким чином: якщо на вхід ЕС поступає стан (a_1, b_3) то, використовуючи відношення r_2 , можна перейти до стану (a_3, b_2) , на основі якого можна ставити діагноз, використовуючи стандартний продукційний висновок.

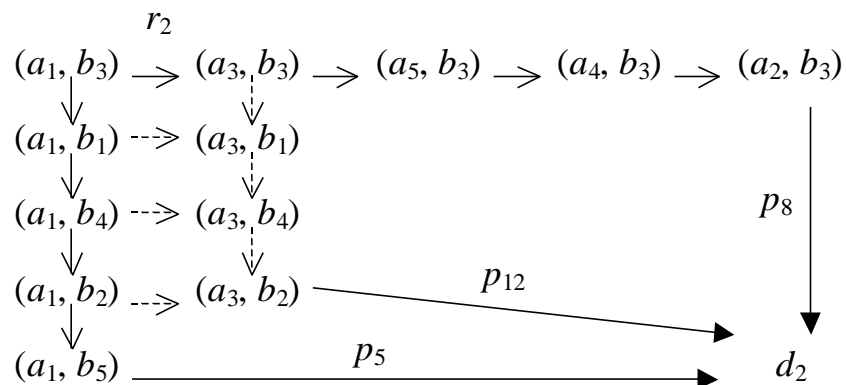


Рисунок 2.17 – Постановка діагнозу d_2

Нехай експерт відповів, що значення a_4 параметра a характерніше для діагнозу d_3 , ніж значення a_2 , яке, у свою чергу, характерніше для діагнозу d_3 , ніж значення a_1 , характерніше для діагнозу d_3 , ніж значення a_3 .

Нехай експерт відповів, що значення b_4 параметра b характерніше для діагнозу d_3 , ніж значення b_2 , яке, у свою чергу, характерніше для діагнозу d_3 , ніж значення b_5 , характерніше для діагнозу d_3 , ніж значення b_3 .

Бінарні відносини r_{a3} , r_{b3} характерності значень параметрів a , b для діагнозу d_3 можна представити графічно так, як показано на рисунку 2.18.

$$\begin{array}{ccc}
 r_{a3} & & r_{b3} \\
 a_4 \longrightarrow a_2 \longrightarrow a_1 \longrightarrow a_3 & & b_4 \longrightarrow b_2 \longrightarrow b_5 \longrightarrow b_3
 \end{array}$$

Рисунок 2.18 – Відносини характерності r_{a3} і r_{b3}

Відносини r_{a3} і r_{b3} дозволяють побудувати бінарне відношення r_3 на множині станів системи S , якій може бути поставлений діагноз d_3 . Це відношення r_3 дозволяє ставити діагноз d_3 тільки за допомогою продукцій p_3 , p_{14} (рисунок 2.19). Постановка діагнозу в ЕС, яка заснована на модифікованій таким чином базі знань KB_4 , відбувається, наприклад, таким чином: якщо на вхід ЕС поступає стан (a_4, b_4) то, використовуючи відношення r_3 , можна перейти до стану (a_1, b_3) , на основі якого можна ставити діагноз, використовуючи стандартний продукційний висновок.

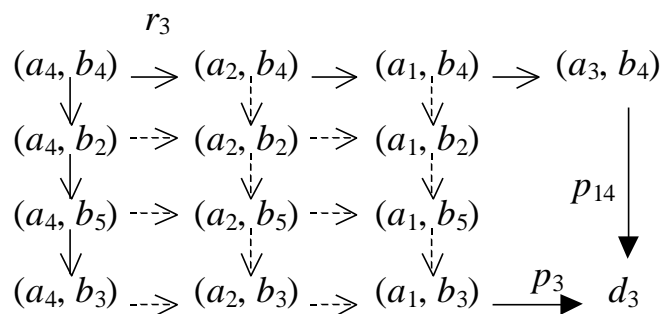


Рисунок 2.19 – Постановка діагнозу d_3

Нехай експерт відповів, що значення a_3 параметра a характерніше для діагнозу d_4 , ніж значення a_5 , яке, у свою чергу, характерніше для діагнозу d_4 , ніж значення a_4 , характерніше для діагнозу d_4 , ніж значення a_2 . Найхарактернішим значенням параметра a для діагнозу d_4 є значення a_1 .

Нехай експерт відповів, що значення b_5 параметра b характерніше для діагнозу d_4 , ніж значення b_3 , яке, у свою чергу, характерніше для діагнозу d_4 , ніж значення b_1 , характерніше для діагнозу d_4 , ніж значення b_4 . Найхарактернішим значенням параметра b для діагнозу d_4 є значення b_2 .

Бінарні відносини r_{a4} , r_{b4} характерності значень параметрів a , b для діагнозу

d_4 можна представити графічно так, як показано на рисунку 2.20.



Рисунок 2.20 – Відносини характерності r_{a4} і r_{b4}

Відносини r_{a4} і r_{b4} дозволяють побудувати бінарне відношення r_4 на множині станів системи S , якій може бути поставлений діагноз d_4 . Це відношення r_4 дозволяє ставити діагноз d_4 тільки за допомогою продукцій p_3 , p_{12} (рисунок 2.21). Постановка діагнозу в ЕС, яка заснована на модифікованій таким чином базі знань KB_4 , відбувається, наприклад, таким чином: якщо на вхід ЕС поступає стан (a_3, b_5) то, використовуючи відношення r_3 , можна перейти до стану (a_1, b_3) , на основі якого можна ставити діагноз, використовуючи стандартний продукційний висновок.

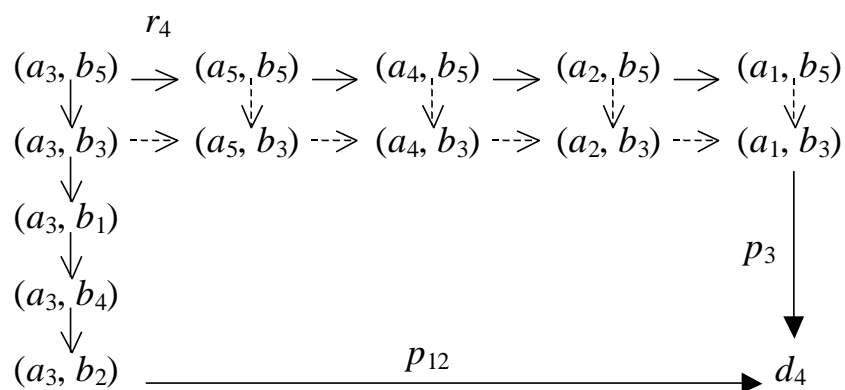


Рисунок 2.21 – Постановка діагнозу d_4

Нехай експерт відповів, що значення a_5 параметра a характерніше для діагнозу d_5 , ніж значення a_4 , яке, у свою чергу, характерніше для діагнозу d_5 , ніж значення a_2 , характерніше для діагнозу d_5 , ніж значення a_1 . Нехай експерт відповів, що значення b_2 параметра b характерніше для діагнозу d_5 , ніж значення b_5 , яке, у свою чергу, характерніше для діагнозу d_5 , ніж значення b_3 , характерніше для діагнозу d_5 , ніж значення b_1 . Найхарактернішим значенням параметра b для діагнозу d_5 є значення b_4 .

Бінарні відносини r_{a4} , r_{b4} характерності значень параметрів a , b для діагнозу d_4 можна представити графічно так, як показано на рисунку 2.22.

$$\begin{array}{ccc}
 r_{a5} & & r_{b5} \\
 a_5 \rightarrow a_4 \rightarrow a_2 \rightarrow a_1 & & b_2 \rightarrow b_5 \rightarrow b_3 \rightarrow b_1 \rightarrow b_4
 \end{array}$$

Рисунок 2.22 – Відносини характерності r_{a5} і r_{b5}

Відносини r_{a5} і r_{b5} дозволяють побудувати бінарне відношення r_5 на множині станів системи S , якій може бути поставлений діагноз d_5 . Це відношення r_5 дозволяє ставити діагноз d_5 тільки за допомогою продукцій p_5 , p_8 і p_{24} (рисунок 2.23). Постановка діагнозу в ЕС, яка заснована на модифікованій таким чином базі знань KB_4 , відбувається, наприклад, таким чином: якщо на вхід ЕС поступає стан (a_5, b_2) то, використовуючи відношення r_3 , можна перейти до стану (a_2, b_3) , на основі якого можна ставити діагноз, використовуючи стандартний продукційний висновок.

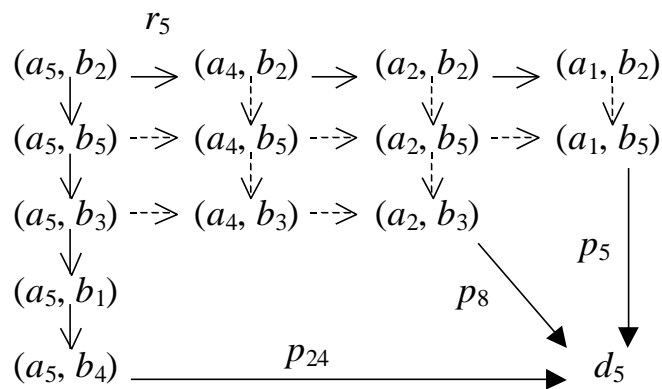


Рисунок 2.23 – Постановка діагнозу d_5

Таким чином, використовуючи бінарні відносини r_1 , r_2 , r_3 , r_4 і r_5 можна залишити в базі знань KB_4 з 25 продукцій тільки 8 продукцій: p_3 , p_5 , p_8 , p_{12} , p_{14} , p_{16} , p_{21} і p_{24} .

Діагностика комп'ютерної техніки з трьома параметрами. Нехай комп'ютерна система S характеризується трьома параметрами a , b і c . Параметр a може приймати три значення: a_1 , a_2 , a_3 . Параметр b може приймати три значення: b_1 , b_2 , b_3 . Параметр c може приймати два значення: c_1 , c_2 . Нехай множина значень діагнозу d , який може бути поставлений системі S , складається з трьох елементів:

- d_0 – система S справна;
- d_1 – у системи S несправність одного типу;

– d_2 – у системи S несправність іншого типу.

База знань KB_5 продукційної експертної системи реального часу, здатна поставити діагноз будь-якому стану системи S , може полягати, наприклад, з наступної множини P продукцій:

- p_1 – «Якщо $a = a_1, b = b_1$ і $c = c_1$, то $d = d_1$ »;
- p_2 – «Якщо $a = a_1, b = b_1$ і $c = c_2$, то $d = d_1$ »;
- p_3 – «Якщо $a = a_1, b = b_2$ і $c = c_1$, то $d = d_0$ »;
- p_4 – «Якщо $a = a_1, b = b_2$ і $c = c_2$, то $d = d_0$ »;
- p_5 – «Якщо $a = a_1, b = b_3$ і $c = c_1$, то $d = d_2$ »;
- p_6 – «Якщо $a = a_1, b = b_3$ і $c = c_2$, то $d = d_2$ »;
- p_7 – «Якщо $a = a_2, b = b_1$ і $c = c_1$, то $d = d_1$ »;
- p_8 – «Якщо $a = a_2, b = b_1$ і $c = c_2$, то $d = d_1$ »;
- p_9 – «Якщо $a = a_2, b = b_2$ і $c = c_1$, то $d = d_2$ »;
- p_{10} – «Якщо $a = a_2, b = b_2$ і $c = c_2$, то $d = d_2$ »;
- p_{11} – «Якщо $a = a_2, b = b_3$ і $c = c_1$, то $d = d_2$ »;
- p_{12} – «Якщо $a = a_2, b = b_3$ і $c = c_2$, то $d = d_2$ »;
- p_{13} – «Якщо $a = a_3, b = b_1$ і $c = c_1$, то $d = d_1$ »;
- p_{14} – «Якщо $a = a_3, b = b_1$ і $c = c_2$, то $d = d_1$ »;
- p_{15} – «Якщо $a = a_3, b = b_2$ і $c = c_1$, то $d = d_2$ »;
- p_{16} – «Якщо $a = a_3, b = b_2$ і $c = c_2$, то $d = d_2$ »;
- p_{17} – «Якщо $a = a_3, b = b_3$ і $c = c_1$, то $d = d_2$ »;
- p_{18} – «Якщо $a = a_3, b = b_3$ і $c = c_2$, то $d = d_2$ ».

Експертові, знання якого зафіксовані в базі знань KB_5 , можна поставити наступні п'ять питань:

- яке значення параметра a характерніше для діагнозу d_1 ;
- яке значення параметра c характерніше для діагнозу d_1 ;
- яке значення параметра a характерніше для діагнозу d_2 ;
- яке значення параметра b характерніше для діагнозу d_2 ;
- яке значення параметра c характерніше для діагнозу d_2 .

Нехай експерт відповів, що значення a_1 параметра a характерніше для діагнозу d_1 , ніж значення a_2 . Найхарактернішим значенням параметра a для

діагнозу d_1 є значення a_3 .

Нехай експерт відповів, що значення c_1 параметра c характерніше для діагнозу d_1 , ніж значення c_2 .

Бінарні відносини r_{a1} , r_{c1} характерності значень параметрів a , c для діагнозу d_1 можна представити графічно так, як показано на рисунку 2.24.

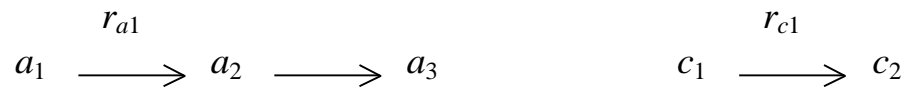


Рисунок 2.24 – Відносини характерності r_{a1} і r_{c1}

Відносини r_{a1} , r_{c1} дозволяють побудувати бінарне відношення r_1 на множині станів системи S , якій може бути поставлений діагноз d_1 . Відношення r_1 дозволяє ставити діагноз d_1 за допомогою тільки однієї продукції p_{14} (рисунок 2.25). Постановка діагнозу в ЕС, яка заснована на модифікованій таким чином базі знань KB_5 , відбувається, наприклад, таким чином: якщо на вхід ЕС поступає стан (a_1, b_1, c_1) то, використовуючи відношення r_1 , можна перейти до стану (a_3, b_1, c_2) , на основі якого можна ставити діагноз, використовуючи стандартний продукційний висновок.

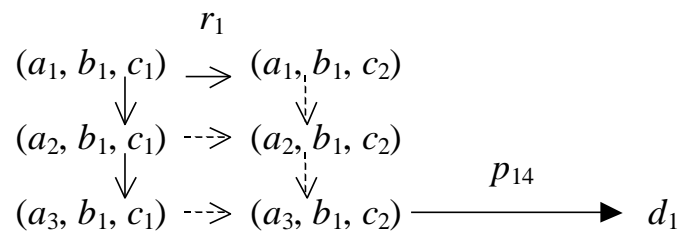


Рисунок 2.25 – Постановка діагнозу d_1

Нехай експерт відповів, що значення a_3 параметра a характерніше для діагнозу d_2 , ніж значення a_2 . Найхарактернішим значенням параметра a для діагнозу d_2 є значення a_1 .

Нехай експерт відповів, що значення b_3 параметра b характерніше для діагнозу d_2 , ніж значення b_2 .

Нехай експерт відповів, що значення c_2 параметра c характерніше для діагнозу d_2 , ніж значення c_1 .

Бінарні відносини r_{a2} , r_{b2} , r_{c2} характерності значень параметрів a , b , c для діагнозу d_2 можна представити графічно так, як показано на рисунку 2.26.

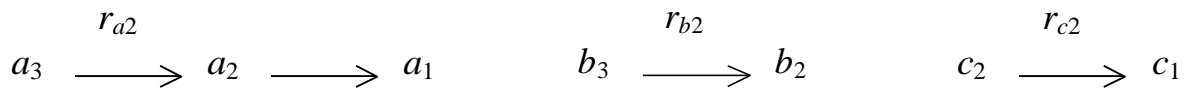


Рисунок 2.26 – Відносини характерності r_{a2} , r_{b2} і r_{c2}

Відносини r_{a2} , r_{b2} і r_{c2} дозволяють побудувати бінарне відношення r_2 на множині станів системи S , якій може бути поставлений діагноз d_2 . Відношення r_2 дозволяє ставити діагноз d_2 тільки за допомогою продукцій p_5 і p_9 (рисунок 2.27). Постановка діагнозу в ЕС, яка заснована на модифікованій таким чином базі знань KB_5 , відбувається, наприклад, таким чином: якщо на вхід ЕС поступає стан (a_3, b_3, c_2) то, використовуючи відношення r_2 , можна перейти до стану (a_1, b_3, c_1) , на основі якого можна ставити діагноз, використовуючи стандартний продукційний висновок.

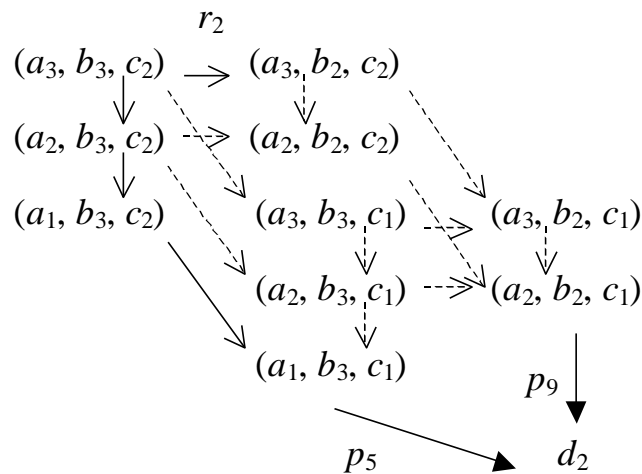


Рисунок 2.27 – Постановка діагнозу d_2

Таким чином, використовуючи бінарні відносини r_1 і r_2 можна залишити в базі знань KB_5 з 18 продукцій тільки 5 продукцій: p_3, p_4, p_5, p_9 і p_{14} .

Діагностика комп'ютерної техніки з багатьма параметрами. Нехай комп'ютерна система S характеризується множиною $A = \{a_n \mid n=1, \dots, N\}$ параметрів, де a_n – n -й параметр системи; S, N – кількість параметрів системи S . Позначимо через $a_n = \{a_{n,m} \mid m=1, \dots, M\}$ множини значень параметра a_n системи S , де $a_{n,m}$ – m -е значення n -

го параметра системи S , $M(n)$ – кількість значень n -го параметра системи S .

Декартовий добуток $C = a_1 \times a_2 \times \dots \times a_n \times \dots \times a_N$ множин значень параметрів комп'ютерної системи S є множиною всіх гіпотетично можливих станів комп'ютерної системи S . Таким чином, стан системи S характеризується вектором $c = \langle c_1, \dots, c_n, \dots, c_N \rangle$ де $c_n \in a_n$.

Нехай $D = \{d_k \mid k=0, \dots, K\}$ – множина діагнозів, які можуть бути поставлені системі S , де d_k – k -й діагноз, який можна поставити системі S , d_0 – «система S справна», d_k – «у системи S несправність k -го типу» ($k = \overline{1, K}$), K – кількість несправностей системи S .

Нехай база знань продукційної експертної системи реального часу, здатної поставити діагноз будь-якому стану системи S , складається з наступної множини $P = \{p_q \mid q=1, \dots, Q\}$ продукцій, де p_q – q -а продукція, Q – кількість продукцій.

Продукція p_q може бути одного з двох типів. Продукція першого типу відображає знання експерта при проміжних міркуваннях в процесі постановки діагнозу. Найпростіший вигляд, який може мати така продукція: «Якщо параметр a_n має значення $a_{n,m}$, то перевірити значення параметра b », де $b \in A \setminus \{a_n\}$.

Складніші продукції першого типу можуть мати вигляд: «Якщо параметр a_i має значення $a_{i,m}$ і параметр a_j має значення $a_{j,n}$, то перевірити значення параметрів b і c », де $b, c \in A \setminus \{a_i, a_j\}$.

Використовуючи символи математичної логіки \neg , \wedge , \vee для позначення відповідно заперечення «НЕ» і зв'язок «І», «АБО», можна записати ще складніші продукції першого типу. Наприклад, «Якщо $a_i = a_{i,m} \wedge a_j = a_{j,n} \vee a_k = a_{k,l}$, то перевірити значення параметрів b , c і e », де $b, c, e \in A \setminus \{a_i, a_j, a_k\}$.

У загальному вигляді продукцію першого типу можна записати таким чином: «Якщо $f_B \langle b_i \rangle = b_{i,m} \langle C \rangle$, то перевірити значення параметрів C », де $b_i = b_{i,m} \langle C \rangle$ – простий вислів «параметр b_i має значення $b_{i,m} \langle C \rangle$ », f – складний вислів (складене з простих, використовуючи дужки і символи \neg , \wedge , \vee) $b_i \in B \subset A \quad C \subset A \setminus B$.

Продукція другого типу відображає знання експерта при постановці

остаточного діагнозу. Найпростіший вигляд, який може мати така продукція: «Якщо параметр a_n має значення $a_{n,m}$, то у системи S несправність k -го типу». Складніші продукції другого типу можуть мати вигляд: «Якщо параметр a_i має значення $a_{i,m}$ і параметр a_j має значення $a_{j,n}$, то d_k ».

Використовуючи символи \neg , \wedge , \vee можна записати ще складніші продукції другого типу. Наприклад, «Якщо $a_i = a_{i,m} \wedge a_j = a_{j,n} \vee a_k = a_{k,l}$, то d_l і d_c ».

У загальному вигляді продукцію другого типу можна записати таким чином: «Якщо $f_B \mathbb{C}_i = b_{i,m} \mathbb{C}$ то E », де $E = \mathbb{C}_0$ або $E = D \setminus \mathbb{C}_0$.

Експертові, знання якого зафіксовані в базі знань, можна поставити наступні питання: яке значення параметра a_n характерніший для діагнозу d_k ($n = \overline{1, N}$; $k = \overline{1, K}$).

Нехай експерт відповів, що значення $a_{n,i}$ параметра a_n характерніше для діагнозу d_k , ніж $a_{n,j}$ ($n = \overline{1, N}$; $k = \overline{1, K}$; $i, j = \overline{1, M} \mathbb{C}$; $ij \neq$). Позначимо це через $a_{n,i}^k > a_{n,j}$.

Відповіді на ці питання дозволяють побудувати бінарне відношення характерності значень параметра a_n для діагнозу d_k : $r_{n,k} = \{(a_{n,i}; a_{n,j}) \mid a_{n,i}^k > a_{n,j}\}$ ($n = \overline{1, N}$; $k = \overline{1, K}$).

На основі відносин $r_{n,k}$ ($n = \overline{1, N}$) можна побудувати бінарне відношення характерності всіх гіпотетично можливих станів комп'ютерної системи S для діагнозу d_k : $r_k = \{(c1; c2) \mid \forall n = \overline{1, N} \ c_n^1 \geq c_n^2; \exists n_0 = \overline{1, N} \ c_{n_0}^1 > c_{n_0}^2\}$.

Нехай C_k – множина станів комп'ютерної системи S , якій можна поставити діагноз d_k ($C_k \subset C$). Розглянемо множину станів, найменш характерних для діагнозу d_k : $C_k^{\min} = \{c \in C_k \mid \forall b \in C_k \ (b;c) \in r_k\}$ ($C_k^{\min} \subset C_k \ k = \overline{1, K}$). Нехай P_k – множина продукцій, за допомогою яких ставиться діагноз d_k складаються з множини C_k^{\min} ($k = \overline{1, K}$).

Отже, будь-якому стану з множини C_k діагноз d_k можна поставити, використовуючи тільки множину продукцій P_k і відношення r_k ($k = \overline{1, K}$).

Розглянемо довільний стан $c \in C_k$. Якщо $c \in C_k^{\min}$, то цьому стану можна поставити діагноз d_k , використовуючи тільки множину продукцій P_k . Якщо $c \in C_k^{\min}$, то, використовуючи відношення r_k , знайдемо такий стан $b \in C_k^{\min}$, що $(c; b) \in r_k$. Стану $b \in C_k^{\min}$ можна поставити діагноз d_k , використовуючи тільки безліч продукцій P_k . Що і потрібно було довести.

Нехай P_0 – множина продукцій, за допомогою яких ставиться діагноз d_0 . Розглянемо множину продукцій $P_{\min} = \bigcup_{k=0}^K P_k$ ($P_{\min} \subset P$) і множина відносин $R = \bigcup_{k=1}^K r_k$.

Будь-якому стану з множини C діагноз s множини D можна поставити, використовуючи тільки множину P_{\min} і R .

Розглянемо довільний стан $c \in C$. Якщо $c \in C_0$, то цьому стану можна поставити діагноз d_0 , використовуючи тільки множину продукцій P_0 . Якщо $c \in C_k$ ($k = \overline{1, K}$), то, згідно леми 2.1, цьому стану можна поставити діагноз d_k , використовуючи тільки множину продукцій $P_k \subset P_{\min}$ і відношення $r_k \in R$ ($k = \overline{1, K}$).

2.2 Метод аналізу бази знань в експертних системах діагностики комп'ютерних систем

Нехай комп'ютерна система S характеризується тільки одним параметром a , який може приймати всього три значення: a_1 , a_2 і a_3 . Нехай множина значень діагнозу d , який може бути поставлений системі S складається з двох елементів:

- d_0 – система S справна;
- d_1 – система S несправна.

База знань KB_1 продукційної експертної системи реального часу, здатна поставити діагноз будь-якому стану системи S , може складатися з деякої множини P продукцій:

Для аналізу цієї множини P продукцій експертів, знання якого зафіксовані

в базі знань KB_1 , можна поставити наступні два питання:

- яке значення параметра a характерніше для діагнозу d_0 ;
- яке значення параметра a характерніше для діагнозу d_1 .

Нехай експерт відповів, що значення a_1 параметра a характерніше для діагнозу d_0 , ніж значення a_2 , яке, у свою чергу, характерніше для діагнозу d_0 , ніж значення a_3 .

Нехай експерт відповів, що значення a_3 параметра a характерніше для діагнозу d_1 , ніж значення a_2 , яке, у свою чергу, характерніше для діагнозу d_1 , ніж значення a_1 .

Ці відповіді експерта дозволяють побудувати бінарне відношення r_0 характерності всіх можливих станів комп'ютерної системи S для діагнозу d_0 і бінарне відношення r_1 характерності всіх можливих станів комп'ютерної системи S для діагнозу d_1 так, як це показано на рисунку 2.28.

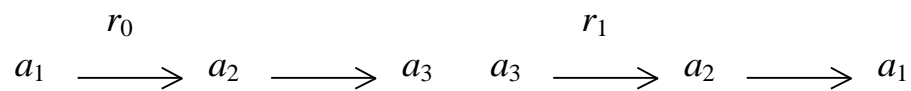


Рисунок 2.28 – Відносини характерності r_0 і r_1

Для аналізу бази знань KB_1 необхідно пред'явити їй для діагностики деякий стан комп'ютерної системи S .

Стани володіють різним ступенем інформативності щодо діагнозу – різною кількістю станів, що побічно діагностуються щодо даного діагнозу при явному пред'явленні стану. Наприклад, якщо стану a_2 буде поставлений діагноз d_0 , то і одному характернішому для діагнозу d_0 стану a_1 також повинен бути поставлений діагноз d_0 . Аналогічно, якщо стану a_2 не буде поставлений діагноз d_0 , то і одному менш характерному для діагнозу d_0 стану a_3 також не повинен бути поставлений діагноз d_0 . Інформативність стану щодо діагнозу залежить від відповіді експерта (бази знань KB_1). У будь-якому випадку можна гарантувати мінімум інформативності з двох можливих відповідей експерта. У даному прикладі – мінімум з 1 і 1. Цей мінімум (1) і можна вважати ступенем інформативності стану a_2 щодо діагнозу d_0 .

Інформативність стану визначається сумою його інформативностей щодо

кожного діагнозу.

Таким чином, для швидкого аналізу бази знань KB_1 необхідно пред'явити їй для діагностики найбільш інформативний в даний момент стан комп'ютерної системи S .

Інформативність всіх станів в даному прикладі показана в таблиці 2.1.

Таблиця 2.1 – Інформативність станів

Стан	Інформативність		
	щодо діагнозу d_0	щодо діагнозу d_1	сума
a_1	0	0	0
a_2	1	1	2
a_3	0	0	0

Для діагностики необхідно вибрати найбільш інформативний стан a_2 .

Нехай в базі знань KB_1 продукційної експертної системи реального часу, здатної поставити діагноз будь-якому стану системи S , є наступна продукція: p_2 – «Якщо $a = a_2$, то $d = d_1$ ».

Результати діагностики стану a_2 показані в таблиці 2.2.

Таблиця 2.2 – Результати діагностики стану a_2

Стан	Діагноз	
	d_0	d_1
a_1		
a_2	–	+
a_3	–	+

Відмітимо, що повністю діагностований стан a_3 .

Графічно результати діагностики стану a_2 можна показати так, як на рисунку 2.29.

Залишився один не діагностований стан a_1 .

Нехай в базі знань KB_1 продукційної експертної системи реального часу,

здатної поставити діагноз будь-якому стану системи S , є наступна продукція: p_1 – «Якщо $a = a_1$, то $d = d_0$ ».

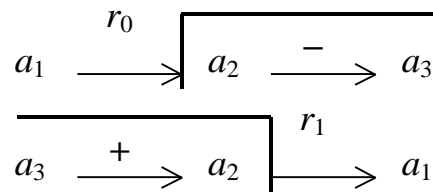


Рисунок 2.29 – Графічний опис результату діагностики стану a_2

Тоді для діагностики будь-якого стану системи S достатньо відносин r_0 і r_1 , а також наступних двох продукцій:

- p_1 – «Якщо $a = a_1$, то $d = d_0$ »;
- p_2 – «Якщо $a = a_2$, то $d = d_1$ ».

Нехай знову комп'ютерна система S характеризується тільки одним параметром a , але який тепер може приймати вже дев'ять значень: $a_1, a_2, a_3, a_4, a_5, a_6, a_7, a_8$ і a_9 . Нехай множина значень діагнозу d , який може бути поставлений системі S , складається з трьох елементів:

- d_0 – система S справна;
- d_1 – у системи S несправність одного типу;
- d_2 – у системи S несправність іншого типу.

База знань KB_2 продукційної експертної системи реального часу, здатна поставити діагноз будь-якому стану системи S , може полягати, з деякої множини P продукцій.

Для аналізу цієї множини P продукцій експертові, знання якого зафіксовані в базі знань KB_2 , можна поставити наступні три питання:

- яке значення параметра a характерніше для діагнозу d_0 ;
- яке значення параметра a характерніше для діагнозу d_1 ;
- яке значення параметра a характерніше для діагнозу d_2 .

Нехай експерт відповів, що значення a_3 параметра a характерніше для діагнозу d_0 , ніж значення a_2 , яке характерніше для діагнозу d_0 , ніж значення a_1 , яке характерніше для діагнозу d_0 , ніж значення a_4 , яке характерніше для діагнозу d_0 , ніж значення a_7 , яке характерніше для діагнозу d_0 , ніж значення a_5 , яке характерніше для діагнозу d_0 , ніж значення a_8 , яке характерніше для діагнозу d_0 ,

ніж значення a_6 , яке характерніше для діагнозу d_0 , ніж значення a_9 .

Нехай експерт відповів, що значення a_6 параметра a характерніше для діагнозу d_1 , ніж значення a_5 , яке характерніше для діагнозу d_1 , ніж значення a_4 , яке характерніший для діагнозу d_1 , ніж значення a_1 , яке характерніший для діагнозу d_1 , ніж значення a_7 , яке характерніший для діагнозу d_1 , ніж значення a_2 , яке характерніший для діагнозу d_1 , ніж значення a_8 , яке характерніший для діагнозу d_1 , ніж значення a_3 , яке характерніший для діагнозу d_1 , ніж значення a_9 .

Нехай експерт відповів, що значення a_9 параметра a характерніший для діагнозу d_2 , ніж значення a_8 , яке характерніше для діагнозу d_2 , ніж значення a_7 , яке характерніше для діагнозу d_2 , ніж значення a_1 , яке характерніше для діагнозу d_2 , ніж значення a_4 , яке характерніше для діагнозу d_2 , ніж значення a_2 , яке характерніше для діагнозу d_2 , ніж значення a_5 , яке характерніше для діагнозу d_2 , ніж значення a_3 , яке характерніше для діагнозу d_2 , ніж значення a_6 .

Ці відповіді експерта дозволяють побудувати бінарне відношення r_0 характерності всіх можливих станів комп'ютерної системи S для діагнозу d_0 , а також бінарне відношення r_1 характерності всіх можливих станів комп'ютерної системи S для діагнозу d_1 , нарешті, бінарне відношення r_2 характерності всіх можливих станів комп'ютерної системи S для діагнозу d_2 так, як це показано на рисунку 2.30.

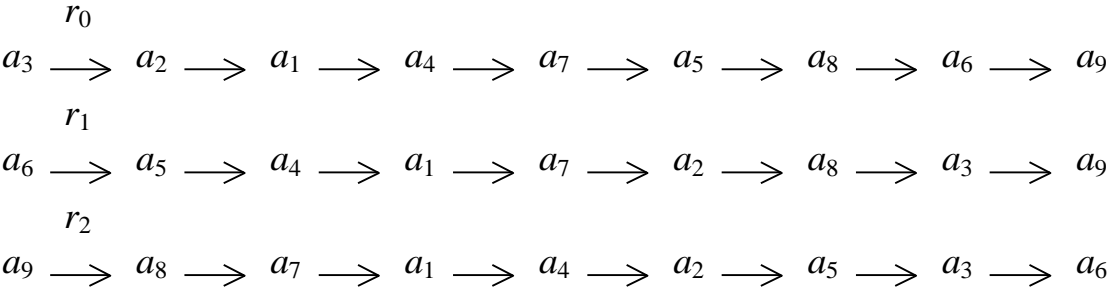


Рисунок 2.30 – Відносини характерності r_0, r_1 і r_2

Для аналізу бази знань KB_2 необхідно пред'явити їй для діагностики деякий стан комп'ютерної системи S .

Стани володіють різним ступенем інформативності щодо діагнозу – різною кількістю станів, що побічно діагностуються щодо даного діагнозу при явному пред'явленні стану. Наприклад, якщо стану a_4 буде поставлений діагноз d_1 , то і

двом характернішим для діагнозу d_1 станам a_5 і a_6 також повинен бути поставлений діагноз d_1 . Аналогічно, якщо стану a_4 не буде поставлений діагноз d_1 , то і шести менш характерним для діагнозу d_1 станам a_1, a_2, a_3, a_7, a_8 і a_9 також не повинен бути поставлений діагноз d_1 . Інформативність стану щодо діагнозу залежить від відповіді експерта (бази знань KB_2). У будь-якому випадку можна гарантувати мінімум інформативності з двох можливих відповідей експерта. У даному прикладі – мінімум з 2 і 6. Цей мінімум (1) і можна вважати ступенем інформативності стану a_4 щодо діагнозу d_1 .

Інформативність стану визначається сумою його інформативностей щодо кожного діагнозу.

Таким чином, для швидкого аналізу бази знань KB_2 необхідно пред'явити їй для діагностики найбільш інформативний в даний момент стан комп'ютерної системи S .

Інформативність всіх станів в даному прикладі показана в таблиці 2.3.

Таблиця 2.3 – Інформативність станів

Стан	Інформативність			
	відн. діагнозу d_0	відн. діагнозу d_1	відн. діагнозу d_2	сума
a_1	2	3	1	6
a_2	1	3	3	7
a_3	0	1	1	2
a_4	3	2	4	9
a_5	3	1	2	6
a_6	1	0	0	1
a_7	4	4	2	10
a_8	2	2	1	5
a_9	0	0	0	0

Для діагностики необхідно вибрати найбільш інформативний стан a_7 .

Нехай в базі знань KB_2 продукційної експертної системи реального часу, яка здатна поставити діагноз будь-якому стану системи S є наступна продукція: p_7 –

«Якщо $a = a_7$, то $d = d_2$ ».

Результати діагностики стану a_7 показані в таблиці 2.4.

Таблиця 2.4 – Результати діагностики стану a_7

Стан	Діагноз		
	d_0	d_1	d_2
a_1			
a_2		–	
a_3		–	
a_4			
a_5	–		
a_6	–		
a_7	–	–	+
a_8	–	–	+
a_9	–	–	+

Відмітимо, що повністю діагностовані стани a_8 і a_9 .

Графічно результати діагностики стану a_7 можна показати так, як на рисунку 2.31.

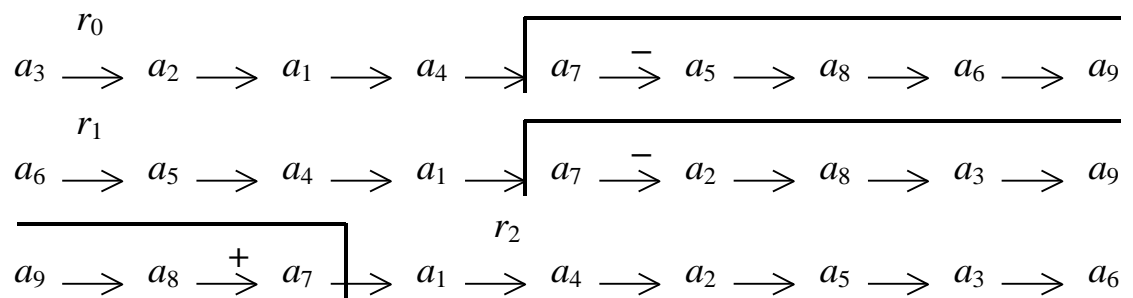


Рисунок 2.31 – Графічний опис результату діагностики стану a_7

Далі необхідно перерахувати інформативність всіх ще не діагностованих станів (таблиця 2.5).

Таблиця 2.5 – Інформативність станів

Стан	Інформативність			
	відн. діагнозу d_0	відн. діагнозу d_1	відн. діагнозу d_2	сума
a_1	1	0	0	1
a_2	1	0	2	3
a_3	0	0	1	1
a_4	0	1	1	2
a_5	0	1	2	3
a_6	0	0	0	0

Для діагностики можна вибрати найбільш інформативний стан a_2 .

Нехай в базі знань KB_2 продукційної експертної системи реального часу, здатної поставити діагноз будь-якому стану системи S є наступна продукція: p_2 – «Якщо $a = a_2$, то $d = d_0$ ».

Результати діагностики стану a_2 показані в таблиці 2.6.

Таблиця 2.6 – Результати діагностики стану a_2

Стан	Діагноз		
	d_0	d_1	d_2
a_1			
a_2	+	–	–
a_3	+	–	–
a_4			
a_5	–		–
a_6	–		–
a_7	–	–	+
a_8	–	–	+
a_9	–	–	+

Відмітимо, що повністю діагностований стан a_3 .

Графічно результати діагностики стану a_2 можна показати так, як на рисунку 2.32.

Далі необхідно перерахувати інформативність всіх ще не діагностованих станів (таблиця 2.7).

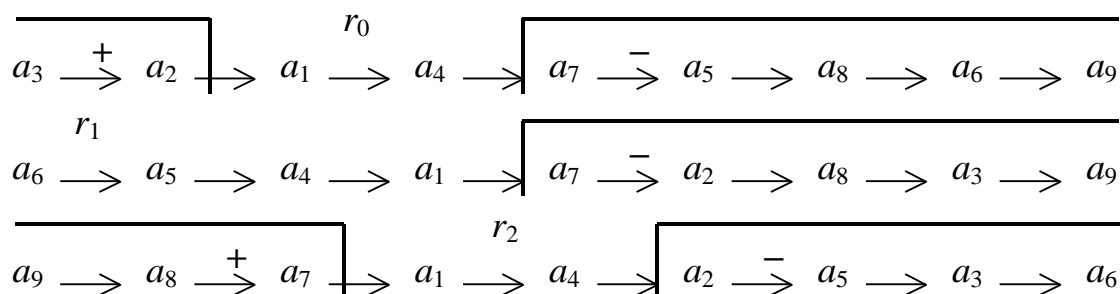


Рисунок 2.32 – Графічний опис результату діагностики стану a_2

Таблиця 2.7 – Інформативність станів

Стан	Інформативність			
	відн. діагнозу d_0	відн. діагнозу d_1	відн. діагнозу d_2	сума
a_1	0	0	0	0
a_4	0	1	0	1
a_5	0	1	0	1
a_6	0	0	0	0

Для діагностики можна вибрати найбільш інформативний стан a_4 .

Нехай в базі знань KB_2 продукційної експертної системи реального часу, здатної поставити діагноз будь-якому стану системи $S \in$ наступна продукція: p_4 – «Якщо $a = a_4$, то $d = d_1$ ».

Результати діагностики стану a_4 показані в таблиці 2.8.

Відмітимо, що повністю діагностовані стани a_5, a_6 .

Графічно результати діагностики стану a_4 можна показати так, як на рисунку 2.33.

Залишився один не діагностований стан a_1 .

Нехай в базі знань KB_2 продукційної експертної системи реального часу, здатної поставити діагноз будь-якому стану системи $S \in$ наступна продукція: p_1 –

«Якщо $a = a_1$, то $d = d_0$ ».

Таблиця 2.8 – Результати діагностики стану a_4

Стан	Діагноз		
	d_0	d_1	d_2
a_1			
a_2	+	-	-
a_3	+	-	-
a_4	-	+	-
a_5	-	+	-
a_6	-	+	-
a_7	-	-	+
a_8	-	-	+
a_9	-	-	+

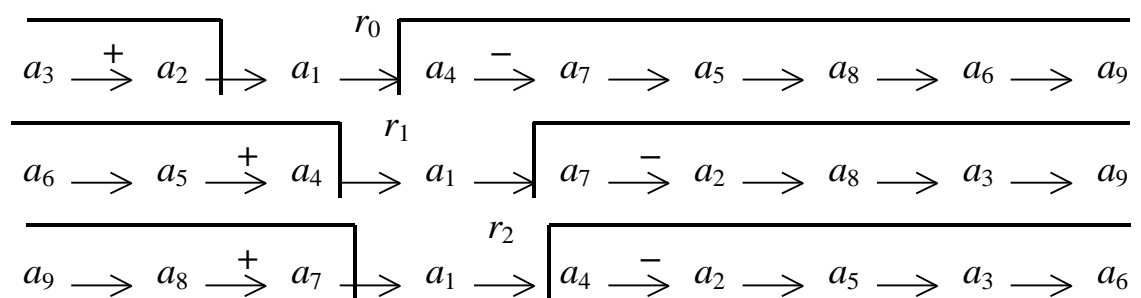


Рисунок 2.33 – Графічний опис результату діагностики стану a_4

Тоді для діагностики будь-якого стану системи S достатньо відносин r_0 , r_1 і r_2 , а також наступних чотири продукції:

- p_7 – «Якщо $a = a_7$, то $d = d_2$ »;
- p_2 – «Якщо $a = a_2$, то $d = d_0$ »;
- p_4 – «Якщо $a = a_4$, то $d = d_1$ »;
- p_1 – «Якщо $a = a_1$, то $d = d_0$ ».

Із списку продукцій також можна виключити продукцію p_2 , оскільки висновок, який можна отримати з її допомогою, повністю забезпечується продукцією p_1 і відношенням r_0 .

В даному розділі дипломної роботи розглянуті продукційні експертні системи, які здатні поставити діагноз будь-якому стану комп'ютерної системи, що характеризуються одним, двома, трьома і довільним числом параметрів. Показано, що використання бінарних відносин характерності значень параметрів для діагнозів дозволяє істотно скоротити число продукцій, необхідних для діагностики довільного стану комп'ютерної системи. Запропонований процес постановки діагнозу, який заснований на продукційному висновку і використанні бінарних відносин характерності значень параметрів для діагнозів. Використовуючи бінарні відносини характерності значень параметрів для діагнозів можна мінімізувати кількість продукцій із збереженням можливості постановки діагнозу будь-якому стану комп'ютерної системи.

3 ВИКОРИСТАННЯ ПРОДУКЦІЙНИХ СИСТЕМ І БІНАРНИХ ВІДНОСИН В ЕКСПЕРТНИХ СИСТЕМАХ ДІАГНОСТИКИ КОМП'ЮТЕРНОЇ ТЕХНІКИ

3.1 Дослідження ефективності використання методів діагностики комп'ютерних систем

Розглянемо приклад процесу побудови формалізованого опису концепцій програмного забезпечення (ПЗ) «Діагноста ПК» на основі представлення знань експерта у вигляді продукційних систем і бінарних відносин.

При пошуку несправностей в ПК використовуються блок-схеми, з яких можна витягувати наступні знання [50].

Нехай множина значень діагнозу d , який може бути поставлений системі S – ПК складається з наступних елементів:

- d_0 – система S справна;
- d_1 – у системи S несправність дисководу гнучких дисків;
- d_2 – у системи S несправність при завантаженні з жорсткого диску;
- d_3 – система S не завантажується;
- d_4 – у системи S при завантаженні – порожній екран;
- d_5 – у системи S несправність клавіатури або миші;
- d_6 – система S працює дуже повільно;
- d_7 – у системи S несправність устаткування або Windows;
- d_8 – система S збоїть;
- d_9 – у системи S дефекти монітора;
- d_{10} – у системи S дефектні зображення монітора;
- d_{11} – у системи S несправність рідкокристалічного монітора;
- d_{12} – у системи S несправність кольору і яскравості монітора;
- d_{13} – у системи S несправність драйверів принтера;
- d_{14} – у системи S несправність принтера;
- d_{15} – у системи S несправність програмного забезпечення принтера;
- d_{16} – у системи S несправність з'єднань;

- d_{17} – у системи S несправність пристроїв CD, CD-R, CD-RW, DVD;
- d_{18} – у системи S несправність пристроїв після інсталяції;
- d_{19} – у системи S несправність змінних дисків;
- d_{20} – система S видає шуми;
- d_{21} – у системи S екран порожній після завантаження.

База знань KB продукційної експертної системи реального часу, яка здатна поставити діагноз d_8 будь-якому стану системи S , може полягати, наприклад, з наступної підмножини продукцій, які відображають знання експерта при постановці остаточного діагнозу:

– $p_{1,1}$ – «Якщо a_1 (значення напруги статичної електрики) = $a_{1,1}$ (напруга_1), то $d = d_8$ »;

– $p_{1,2}$ – «Якщо a_1 (значення напруги статичної електрики) = $a_{1,2}$ (напруга_2), то $d = d_8$ »;

– $p_{1,3}$ – «Якщо a_1 (значення напруги статичної електрики) = $a_{1,3}$ (напруга_3), то $d = d_8$ ».

Конкретні числові характеристики величин «напруга_1», «напруга_2» і «напруга_3» залежать виключно від досвіду експерта і різні експерти можуть задати різні значення цієї напруги.

У експерта також можна з'ясувати, які значення може приймати параметр a_1 . Нехай $a_1 = \{a_{1,1}, a_{1,2}, a_{1,3}, a_{1,4}, a_{1,5}, a_{1,6}\}$, де $a_{1,i}$ – деякий діапазон значень напруги у вольтах (В). Потім експерт може визначити відношення $r_{1,8}$ характерності значень статичної напруги для збоїв комп'ютера: $a_{1,2} \text{ В} > a_{1,3} \text{ В} > a_{1,1} \text{ В} > a_{1,6} \text{ В} > a_{1,4} \text{ В} > a_{1,5} \text{ В}$. Нарешті, експерт може сказати, що $a_{1,6} \text{ В}$ – найменш характерне значення статичної напруги, при якій комп'ютер ще збоїтиме, але при значенні $a_{1,4} \text{ В}$ комп'ютер збоїти не буде.

З цього виходить, що в базу знань KB необхідно додати продукцію $p_{1,6}$ – «Якщо a_1 (значення напруги статичної електрики) = $a_{1,6}$ (напруга_6), то $d = d_8$ ».

При цьому продукції $p_{1,1}$, $p_{1,2}$ і $p_{1,3}$ можна замінити відношенням $r_{1,8}$.

База знань KB продукційної експертної системи реального часу, яка здатна поставити діагноз d_8 будь-якому стану системи S , може містити також наступну підмножину продукцій, які відображають знання експерта при постановці

остаточного діагнозу:

– $p_{2,1}$ – «Якщо a_2 (значення температури всередині комп'ютера) = $a_{2,1}$ (температура_1), то $d = d_8$ »;

– $p_{2,2}$ – «Якщо a_2 (значення температури всередині комп'ютера) = $a_{2,2}$ (температура_2), то $d = d_8$ »;

– $p_{2,3}$ – «Якщо a_2 (значення температури всередині комп'ютера) = $a_{2,3}$ (температура_3), то $d = d_8$ »;

– $p_{2,4}$ – «Якщо a_2 (значення температури всередині комп'ютера) = $a_{2,1}$ (температура_4), то $d = d_8$ ».

Конкретні числові характеристики величин «температура_1», «температура_2», «температура_3» і «температура_4» залежать виключно від досвіду експерта, і різні експерти можуть задати різні значення цих температур.

У експерта також можна з'ясувати, які значення може приймати параметр a_2 . Нехай $a_2 = \{a_{2,1}, a_{2,2}, a_{2,3}, a_{2,4}, a_{2,5}, a_{2,6}, a_{2,7}\}$, де $a_{2,i}$ – деякий діапазон значень температур в градусах Цельсія ($^{\circ}\text{C}$). Потім експерт може визначити відношення $r_{2,8}$ характерності значень температури усередині комп'ютера для його збоїв: $a_{2,4}^{\circ}\text{C} > a_{2,3}^{\circ}\text{C} > a_{2,1}^{\circ}\text{C} > a_{2,2}^{\circ}\text{C} > a_{2,5}^{\circ}\text{C} > a_{2,6}^{\circ}\text{C} > a_{2,7}^{\circ}\text{C}$. Нарешті, експерт може сказати, що $a_{2,5}^{\circ}\text{C}$ – найменш характерне значення температури усередині комп'ютера, при якому він ще збоїтиме, але при значенні $a_{2,6}^{\circ}\text{C}$ комп'ютер збоїти не буде.

З цього виходить, що в базу знань *KB* необхідно додати продукцію $p_{2,5}$ – «Якщо a_2 (значення температури всередині комп'ютера) = $a_{2,5}$ (температура_5), то $d = d_8$ ».

При цьому продукції $p_{2,1}, p_{2,2}, p_{2,3}$ і $p_{2,4}$ можна замінити відношенням $r_{2,8}$.

База знань *KB* продукційної експертної системи реального часу, яка здатна поставити діагноз d_8 будь-якому стану системи S , може містити також наступну підмножину продукцій, які відображають знання експерта при постановці остаточного діагнозу:

– $p_{3,1}$ – «Якщо a_3 (встановлено нове устаткування) = $a_{3,1}$ (устаткування_1), то $d = d_8$ »;

– $p_{3,2}$ – «Якщо a_3 (встановлено нове устаткування) = $a_{3,2}$ (устаткування_2), то

$d = d_8$ »;

– $p_{3,3}$ – «Якщо a_3 (встановлено нове устаткування) = $a_{3,3}$ (устаткування_3), то

$d = d_8$ »;

– $p_{3,4}$ – «Якщо a_3 (встановлено нове устаткування) = $a_{3,4}$ (устаткування_4), то

$d = d_8$ »;

– $p_{3,5}$ – «Якщо a_3 (встановлено нове устаткування) = $a_{3,5}$ (устаткування_5), то

$d = d_8$ ».

Конкретні значення величин «устаткування_1», «устаткування_2», «устаткування_3», «устаткування_4» і «устаткування_5» залежать виключно від досвіду експерта, і різні експерти можуть задати різні назви устаткування. Наприклад, це можуть бути модеми, дискові накопичувачі, пристрої читання компакт-дисків, дисководи DVD, принтери, мережеві адаптери, клавіатури, плати відеоадаптерів, монітори, ігрові контролери, пристрої USB (наприклад, телефон), пристрої IEEE 1394 (наприклад, відеокамера) [51].

У експерта також можна з'ясувати, які значення може приймати параметр a_3 . Нехай $a_3 = \{a_{3,1}, a_{3,2}, a_{3,3}, a_{3,4}, a_{3,5}, a_{3,6}, a_{3,7}, a_{3,8}\}$, де $a_{3,i}$ – деяке нове устаткування. Потім експерт може визначити відношення $r_{3,8}$ характерності встановленого нового устаткування для збоїв комп'ютера: $a_{3,4} > a_{3,3} > a_{3,1} > a_{3,2} > a_{3,5} > a_{3,6} > a_{3,7} > a_{3,8}$. Нарешті, експерт може сказати, що $a_{3,7}$ – найменш характерне нове устаткування, при якому комп'ютер ще збоїтиме, але при значенні $a_{3,8}$ комп'ютер збоїти не буде.

З цього виходить, що в базу знань *KB* необхідно додати продукцію $p_{3,7}$ – «Якщо a_3 (встановлено нове устаткування) = $a_{3,7}$ (устаткування_7), то $d = d_8$ ». При цьому продукції $p_{3,1}, p_{3,2}, p_{3,3}, p_{3,4}, p_{3,5}$ можна замінити відношенням $r_{3,8}$.

База знань *KB* продукційної експертної системи реального часу, яка здатна поставити діагноз d_8 будь-якому стану системи S , може містити також наступну підмножину продукцій, які відображають знання експерта при постановці остаточного діагнозу:

– $p_{4,1}$ – «Якщо a_4 (встановлене нове ПЗ) = $a_{4,1}$ (ПЗ_1), то $d = d_8$ »;

– $p_{4,2}$ – «Якщо a_4 (встановлене нове ПЗ) = $a_{4,2}$ (ПЗ_2), то $d = d_8$ »;

– $p_{4,3}$ – «Якщо a_4 (встановлене нове ПЗ) = $a_{4,3}$ (ПЗ_3), то $d = d_8$ »;

- $p_{4,4}$ – «Якщо a_4 (встановлене нове ПЗ) = $a_{4,4}$ (ПЗ_4), то $d = d_8$ »;
- $p_{4,5}$ – «Якщо a_4 (встановлене нове ПЗ) = $a_{4,5}$ (ПЗ_5), то $d = d_8$ »;
- $p_{4,6}$ – «Якщо a_4 (встановлене нове ПЗ) = $a_{4,6}$ (ПЗ_6), то $d = d_8$ ».

Конкретні значення величин «ПЗ_1», «ПЗ_2», «ПЗ_3», «ПЗ_4», «ПЗ_5» і «ПЗ_6» залежать виключно від досвіду експерта, і різні експерти можуть задати різні назви ПЗ. Наприклад, це можуть бути текстові редактори, табличні процесори, системи управління базами даних, графічні редактори, музичні редактори, відео редактори, ігрові програми, системи автоматизованого проектування, програми для наукових розрахунків, системи візуального об'єктно-орієнтованого програмування, антивіруси, додатки для Інтернет.

У експерта також можна з'ясувати, які значення може приймати параметр a_4 . Нехай $a_4 = \{a_{4,1}, a_{4,2}, a_{4,3}, a_{4,4}, a_{4,5}, a_{4,6}, a_{4,7}, a_{4,8}, a_{4,9}\}$, де $a_{4,i}$ – деяке нове ПЗ. Потім експерт може визначити відношення $r_{4,8}$ характерності встановленого нового ПЗ для збоїв комп'ютера: $a_{4,2} > a_{4,4} > a_{4,6} > a_{4,1} > a_{4,3} > a_{4,5} > a_{4,7} > a_{4,8} > a_{4,9}$. Нарешті, експерт може сказати, що $a_{4,8}$ – найменш характерне нове ПЗ, при якому комп'ютер ще збоїтиме, але при значенні $a_{4,9}$ комп'ютер збоїти не буде.

З цього виходить, що в базу знань *KB* необхідно додати продукцію $p_{4,8}$ – «Якщо a_4 (встановлене нове ПЗ) = $a_{4,8}$ (ПЗ_8), то $d = d_8$ ». При цьому продукції $p_{4,1}$, $p_{4,2}$, $p_{4,3}$, $p_{4,4}$, $p_{4,5}$, $p_{4,6}$ можна замінити відношенням $r_{4,8}$.

База знань *KB* продукційної експертної системи реального часу, яка здатна поставити діагноз d_8 будь-якому стану системи S , може містити також наступну підмножину продукцій, які відображають знання експерта при постановці остаточного діагнозу:

- $p_{5,1}$ – «Якщо a_5 (управління живленням) = $a_{5,1}$ (схема_1), то $d = d_8$ »;
- $p_{5,2}$ – «Якщо a_5 (управління живленням) = $a_{5,2}$ (схема_2), то $d = d_8$ »;
- $p_{5,3}$ – «Якщо a_5 (управління живленням) = $a_{5,3}$ (схема_3), то $d = d_8$ ».

Конкретні значення величин «схема_1», «схема_2» і «схема_3» залежать виключно від досвіду експерта і різні експерти можуть задати різні схеми. Наприклад, залежно від устаткування є можливість автоматичного відключення монітора і жорстких дисків для економії електроенергії, переведення комп'ютера при його простой в режим очікування або в сплячий режим.

У експерта також можна з'ясувати, які значення може приймати параметр a_5 . Нехай $a_5 = \{a_{5,1}, a_{5,2}, a_{5,3}, a_{5,4}, a_{5,5}\}$, де $a_{5,i}$ – деяка схема управління живленням. Потім експерт може визначити відношення $r_{5,8}$ характерності використовуваної схеми управління живленням для збоїв комп'ютера: $a_{5,3} > a_{5,2} > a_{5,1} > a_{5,5} > a_{5,4}$. Нарешті, експерт може сказати, що $a_{5,5}$ – найменш характерна схема управління живленням, при якій комп'ютер ще збоїтиме, але при значенні $a_{5,4}$ комп'ютер збоїти не буде.

З цього виходить, що в базу знань *KB* необхідно додати продукцію $p_{5,5}$ – «Якщо a_5 (управління живленням) = $a_{5,5}$ (схема_5), то $d = d_8$ ». При цьому продукції $p_{5,1}$, $p_{5,2}$ і $p_{5,3}$ можна замінити відношенням $r_{5,8}$.

База знань *KB* продукційної експертної системи реального часу, яка здатна поставити діагноз d_8 будь-якому стану системи S , може містити також наступну підмножину продукцій, які відображають знання експерта при постановці остаточного діагнозу:

- $p_{6,1}$ – «Якщо a_6 (використання ресурсів) = $a_{6,1}$ (режим_1), то $d = d_8$ »;
- $p_{6,2}$ – «Якщо a_6 (використання ресурсів) = $a_{6,2}$ (режим_2), то $d = d_8$ »;
- $p_{6,3}$ – «Якщо a_6 (використання ресурсів) = $a_{6,3}$ (режим_3), то $d = d_8$ »;
- $p_{6,4}$ – «Якщо a_6 (використання ресурсів) = $a_{6,4}$ (режим_4), то $d = d_8$ »;
- $p_{6,5}$ – «Якщо a_6 (використання ресурсів) = $a_{6,5}$ (режим_5), то $d = d_8$ ».

Конкретні значення величин «режим_1», «режим_2», «режим_3», «режим_4» і «режим_5» залежать виключно від досвіду експерта і різні експерти можуть задати різні режими. Наприклад, Windows виділяє системні ресурси відповідно до параметрів настройки і відповідно керує роботою пристроїв. Продуктивність комп'ютера можна підвищити, зокрема, шляхом зміни режиму використання ресурсів часу процесора і пам'яті системою Windows.

У експерта також можна з'ясувати, які значення може приймати параметр a_6 . Нехай $a_6 = \{a_{6,1}, a_{6,2}, a_{6,3}, a_{6,4}, a_{6,5}, a_{6,6}, a_{6,7}\}$, де $a_{6,i}$ – деякий режим використання ресурсів. Потім експерт може визначити відношення $r_{6,8}$ характерності використовуваного режиму використання ресурсів для збоїв комп'ютера: $a_{6,1} > a_{6,2} > a_{6,3} > a_{6,5} > a_{6,4} > a_{6,6} > a_{6,7}$. Нарешті, експерт може сказати, що $a_{6,6}$ – найменш характерний режим використання ресурсів, при якому комп'ютер ще збоїтиме, але

при значенні $a_{6,7}$ комп'ютер збоїти не буде.

З цього виходить, що в базу знань *KB* необхідно додати продукцію $p_{6,6}$ – «Якщо a_6 (використання ресурсів) = $a_{6,6}$ (режим_6), то $d = d_8$ ». При цьому продукції $p_{6,1}, p_{6,2}, p_{6,3}, p_{6,4}$ і $p_{6,5}$ можна замінити відношенням $r_{6,8}$.

База знань *KB* продукційної експертної системи реального часу, яка здатна поставити діагноз d_8 будь-якому стану системи S , може полягати, наприклад, з наступної підмножини продукцій, які відображають знання експерта при постановці остаточного діагнозу:

– $p_{7,1}$ – «Якщо a_7 (значення параметрів електроживлення) = $a_{7,1}$ (параметри_1), то $d = d_8$ »;

– $p_{7,2}$ – «Якщо a_7 (значення параметрів електроживлення) = $a_{7,2}$ (параметри_2), то $d = d_8$ »;

– $p_{7,3}$ – «Якщо a_7 (значення параметрів електроживлення) = $a_{7,3}$ (параметри_3), то $d = d_8$ ».

Конкретні числові характеристики величин «параметри_1», «параметри_2» і «параметри_3» залежать виключно від досвіду експерта і різні експерти можуть задати різні значення цих параметрів. Наприклад, причиною збою комп'ютера може бути інтерференція від різних джерел, перемикання на лініях електропередач, віддалені розряди блискавки, гойдання від вітру на лініях електропередач. Також причиною комп'ютерного збою може бути зниження напруги, викликане наявністю інших електроприладів в схемі. Нарешті, в пікові періоди електростанція може понизити навантаження або взагалі відключити електрику.

У експерта також можна з'ясувати, які значення може приймати a_7 . Нехай $a_7 = \{a_{7,1}, a_{7,2}, a_{7,3}, a_{7,4}, a_{7,5}, a_{7,6}\}$, де $a_{7,i}$ – деякі значення параметрів електроживлення. Потім експерт може визначити відношення $r_{7,8}$ характерності значень параметрів електроживлення для збоїв комп'ютера: $a_{7,2} > a_{7,3} > a_{7,1} > a_{7,6} > a_{7,4} > a_{7,5}$. Нарешті, експерт може сказати, що $a_{7,6}$ – найменш характерне значення параметрів електроживлення, при якому комп'ютер ще збоїтиме, але при значенні $a_{7,4}$ комп'ютер збоїти не буде.

З цього виходить, що в базу знань *KB* необхідно додати продукцію $p_{7,6}$ –

«Якщо a_7 (значення параметрів електроживлення) = $a_{7,6}$ (параметри_6), то $d = d_8$ ».

При цьому продукції $p_{7,1}$, $p_{7,2}$ і $p_{7,3}$ можна замінити відношенням $r_{7,8}$.

База знань *KB* продукційної експертної системи реального часу, яка здатна поставити діагноз будь-якому стану системи S , може складатися також з наступної підмножини продукцій, які відображають знання експерта при проміжних міркуваннях в процесі постановки діагнозу:

– $p_{8,1}$ – «Якщо a_8 (параметри заземлення) = $a_{8,1}$ (параметри_1), то перевірити a_1 »;

– $p_{8,2}$ – «Якщо a_8 (параметри заземлення) = $a_{8,2}$ (параметри_2), то перевірити a_1 »;

– $p_{8,3}$ – «Якщо a_8 (параметри заземлення) = $a_{8,3}$ (параметри_3), то перевірити a_1 »;

– $p_{8,4}$ – «Якщо a_8 (параметри заземлення) = $a_{8,4}$ (параметри_4), то перевірити a_1 ».

Конкретні числові характеристики величин «параметри_1», «параметри_2», «параметри_3» і «параметри_4» залежать виключно від досвіду експерта, і різні експерти можуть задати різні значення цих параметрів.

У експерта також можна з'ясувати, які значення може приймати a_8 . Нехай $a_8 = \{a_{8,1}, a_{8,2}, a_{8,3}, a_{8,4}, a_{8,5}, a_{8,6}, a_{8,7}\}$, де $a_{8,i}$ – деякі значення параметрів заземлення. Потім експерт може визначити відношення $r_{8,8}$ характерності значень параметрів заземлення для збоїв комп'ютера: $a_{8,4} > a_{8,3} > a_{8,1} > a_{8,2} > a_{8,5} > a_{8,6} > a_{8,7}$. Нарешті, експерт може сказати, що $a_{8,5}$ – найменш характерне значення параметрів заземлення, при якому комп'ютер ще збоїтиме, але при значенні $a_{8,6}$ комп'ютер збоїти не буде.

З цього виходить, що в базу знань *KB* необхідно додати продукцію $p_{8,5}$ – «Якщо a_8 (параметри заземлення) = $a_{8,5}$ (параметри_5), то перевірити a_1 ».

При цьому продукції $p_{8,1}$, $p_{8,2}$, $p_{8,3}$ і $p_{8,4}$ можна замінити відношенням $r_{8,8}$.

База знань *KB* продукційної експертної системи реального часу, яка здатна поставити діагноз будь-якому стану системи S , може складатися також з наступної підмножини продукцій, які відображають знання експерта при проміжних міркуваннях в процесі постановки діагнозу:

– $p_{9,1}$ – «Якщо a_9 (вологість повітря в приміщенні) = $a_{9,1}$ (вологість_1), то перевірити a_1 »;

– $p_{9,2}$ – «Якщо a_9 (вологість повітря в приміщенні) = $a_{9,2}$ (вологість_2), то перевірити a_1 »;

– $p_{9,3}$ – «Якщо a_9 (вологість повітря в приміщенні) = $a_{9,3}$ (вологість_3), то перевірити a_1 »;

– $p_{9,4}$ – «Якщо a_9 (вологість повітря в приміщенні) = $a_{9,4}$ (вологість_4), то перевірити a_1 »;

– $p_{9,5}$ – «Якщо a_9 (вологість повітря в приміщенні) = $a_{9,5}$ (вологість_5), то перевірити a_1 ».

Конкретні значення величин «вологість_1», «вологість_2», «вологість_3», «вологість_4» і «вологість_5» залежать виключно від досвіду експерта і різні експерти можуть задати різні величини вологості.

У експерта також можна з'ясувати, які значення може приймати параметр a_9 . Нехай $a_9 = \{a_{9,1}, a_{9,2}, a_{9,3}, a_{9,4}, a_{9,5}, a_{9,6}, a_{9,7}, a_{9,8}\}$, де $a_{9,i}$ – деяке значення вологості. Потім експерт може визначити відношення $r_{9,8}$ характерності вологості повітря в приміщенні для збоїв комп'ютера: $a_{9,4} > a_{9,3} > a_{9,1} > a_{9,2} > a_{9,5} > a_{9,6} > a_{9,7} > a_{9,8}$. Нарешті, експерт може сказати, що $a_{9,7}$ – найменш характерне значення вологості повітря в приміщенні, при якому комп'ютер ще збоїтиме, але при значенні $a_{9,8}$ комп'ютер збоїти не буде.

З цього виходить, що в базу знань *KB* необхідно додати продукцію $p_{9,7}$ – «Якщо a_9 (вологість повітря в приміщенні) = $a_{9,7}$ (вологість_7), то перевірити a_1 ».

При цьому продукції $p_{9,1}, p_{9,2}, p_{9,3}, p_{9,4}, p_{9,5}$ можна замінити відношенням $r_{9,8}$.

База знань *KB* продукційної експертної системи реального часу, яка здатна поставити діагноз будь-якому стану системи S , може складатися також з наступної підмножини продукцій, які відображають знання експерта при проміжних міркуваннях в процесі постановки діагнозу:

– $p_{10,1}$ – «Якщо a_{10} (статична електрика користувача) = $a_{10,1}$ (напруга_1), то перевірити a_1 »;

– $p_{10,2}$ – «Якщо a_{10} (статична електрика користувача) = $a_{10,2}$ (напруга_2), то перевірити a_1 »;

– $p_{10,3}$ – «Якщо a_{10} (статична електрика користувача) = $a_{10,3}$ (напруга_3), то перевірити a_1 »;

– $p_{10,4}$ – «Якщо a_{10} (статична електрика користувача) = $a_{10,4}$ (напруга_4), то перевірити a_1 »;

– $p_{10,5}$ – «Якщо a_{10} (статична електрика користувача) = $a_{10,5}$ (напруга_5), то перевірити a_1 »;

– $p_{10,6}$ – «Якщо a_{10} (статична електрика користувача) = $a_{10,6}$ (напруга_6), то перевірити a_1 ».

Конкретні значення величин «напруга_1» – «напруга_6» залежать виключно від досвіду експерта і різні експерти можуть задати різні значення напруги.

У експерта також можна з'ясувати, які значення може приймати параметр a_{10} . Нехай $a_{10} = \{a_{10,1}, a_{10,2}, a_{10,3}, a_{10,4}, a_{10,5}, a_{10,6}, a_{10,7}, a_{10,8}, a_{10,9}\}$, де $a_{10,i}$ – деяке значення напруги. Потім експерт може визначити відношення $r_{10,8}$ характерності значень статична електрика користувача для збоїв комп'ютера: $a_{10,2} > a_{10,4} > a_{10,6} > a_{10,1} > a_{10,3} > a_{10,5} > a_{10,7} > a_{10,8} > a_{10,9}$. Нарешті, експерт може сказати, що $a_{10,8}$ – найменш характерне значення напруги статичної електрики користувача, при якому комп'ютер ще збоїтиме, але при значенні $a_{10,9}$ комп'ютер збоїти не буде.

З цього виходить, що в базу знань *KB* необхідно додати продукцію $p_{10,8}$ – «Якщо a_{10} (статична електрика користувача) = $a_{10,8}$ (напруга_8), то перевірити a_1 ». При цьому продукції $p_{10,1}, p_{10,2}, p_{10,3}, p_{10,4}, p_{10,5}, p_{10,6}$ можна замінити відношенням $r_{10,8}$.

База знань *KB* продукційної експертної системи реального часу, яка здатна поставити діагноз будь-якому стану системи S , може складатися також з наступної підмножини продукцій, які відображають знання експерта при проміжних міркуваннях в процесі постановки діагнозу:

– $p_{11,1}$ – «Якщо a_{11} (потік повітря через комп'ютер) = $a_{11,1}$ (швидкість_1), то перевірити a_2 »;

– $p_{11,2}$ – «Якщо a_{11} (потік повітря через комп'ютер) = $a_{11,2}$ (швидкість_2), то перевірити a_2 »;

– $p_{11,3}$ – «Якщо a_{11} (потік повітря через комп'ютер) = $a_{11,3}$ (швидкість_3), то перевірити a_2 ».

Конкретні значення величин «швидкість_1», «швидкість_2» і «швидкість_3» залежать виключно від досвіду експерта і різні експерти можуть задати різні швидкості.

У експерта також можна з'ясувати, які значення може приймати параметр a_{11} . Нехай $a_{11} = \{a_{11,1}, a_{11,2}, a_{11,3}, a_{11,4}, a_{11,5}\}$, де $a_{11,i}$ – деяка швидкість потоку повітря через комп'ютер. Потім експерт може визначити відношення $r_{11,8}$ характерності швидкостей потоку повітря через комп'ютер для збоїв комп'ютера: $a_{11,3} > a_{11,2} > a_{11,1} > a_{11,5} > a_{11,4}$. Нарешті, експерт може сказати, що $a_{11,5}$ – найменш характерна швидкість потоку повітря, при якій комп'ютер ще збоїти, але при значенні $a_{11,4}$ комп'ютер збоїти не буде.

З цього виходить, що в базу знань *KB* необхідно додати продукцію $p_{11,5}$ – «Якщо a_{11} (потік повітря через комп'ютер) = $a_{11,5}$ (швидкість_5), то перевірити a_2 ». При цьому продукції $p_{11,1}$, $p_{11,2}$ і $p_{11,3}$ можна замінити відношенням $r_{11,8}$.

База знань *KB* продукційної експертної системи реального часу, яка здатна поставити діагноз будь-якому стану системи S , може складатися також з наступної підмножини продукцій, які відображають знання експерта при проміжних міркуваннях в процесі постановки діагнозу:

- $p_{12,1}$ – «Якщо a_{12} (простір для комп'ютера) = $a_{12,1}$ (об'єм_1), то перевірити a_2 »;
- $p_{12,2}$ – «Якщо a_{12} (простір для комп'ютера) = $a_{12,2}$ (об'єм_2), то перевірити a_2 »;
- $p_{12,3}$ – «Якщо a_{12} (простір для комп'ютера) = $a_{12,3}$ (об'єм_3), то перевірити a_2 »;
- $p_{12,4}$ – «Якщо a_{12} (простір для комп'ютера) = $a_{12,4}$ (об'єм_4), то перевірити a_2 »;
- $p_{12,5}$ – «Якщо a_{12} (простір для комп'ютера) = $a_{12,5}$ (об'єм_5), то перевірити a_2 ».

Конкретні значення величин «об'єм_1» – «об'єм_5» залежать виключно від досвіду експерта, і різні експерти можуть задати різні об'єми.

У експерта також можна з'ясувати, які значення може приймати параметр a_{12} . Нехай $a_{12} = \{a_{12,1}, a_{12,2}, a_{12,3}, a_{12,4}, a_{12,5}, a_{12,6}, a_{12,7}\}$, де $a_{12,i}$ – деякий об'єм

простору для комп'ютера. Потім експерт може визначити відношення $r_{12,8}$ характерності використовуваного об'єму для збоїв комп'ютера: $a_{12,1} > a_{12,2} > a_{12,3} > a_{12,5} > a_{12,4} > a_{12,6} > a_{12,7}$. Нарешті, експерт може сказати, що $a_{12,6}$ – найменш характерний об'єм, при якому комп'ютер ще збоїтиме, але при значенні $a_{12,7}$ комп'ютер збоїти не буде.

З цього виходить, що в базу знань *KB* необхідно додати продукцію $p_{12,6}$ – «Якщо a_{12} (простір для комп'ютера) = $a_{12,6}$ (об'єм_6), то перевірити a_2 ». При цьому продукції $p_{12,1}, p_{12,2}, p_{12,3}, p_{12,4}$ і $p_{12,5}$ можна замінити відношенням $r_{12,8}$.

База знань *KB* продукційної експертної системи реального часу, яка здатна поставити діагноз будь-якому стану системи S , може складатися також з наступної підмножини продукцій, які відображають знання експерта при проміжних міркуваннях в процесі постановки діагнозу:

– $p_{13,1}$ – «Якщо a_{13} (температура в приміщенні) = $a_{13,1}$ (температура_1), то перевірити a_2 »;

– $p_{13,2}$ – «Якщо a_{13} (температура в приміщенні) = $a_{13,2}$ (температура_2), то перевірити a_2 »;

– $p_{13,3}$ – «Якщо a_{13} (температура в приміщенні) = $a_{13,3}$ (температура_3), то перевірити a_2 ».

Конкретні числові характеристики величин «температура_1», «температура_2» і «температура_3» залежать виключно від досвіду експерта, і різні експерти можуть задати різні значення цих температур.

У експерта також можна з'ясувати, які значення може приймати a_{13} . Нехай $a_{13} = \{a_{13,1}, a_{13,2}, a_{13,3}, a_{13,4}, a_{13,5}, a_{13,6}\}$, де $a_{13,i}$ – деяке значення температури в приміщенні. Потім експерт може визначити відношення $r_{13,8}$ характерності значень температури в приміщенні для збоїв комп'ютера: $a_{13,2} > a_{13,3} > a_{13,1} > a_{13,6} > a_{13,4} > a_{13,5}$. Нарешті, експерт може сказати, що $a_{13,6}$ – найменш характерне значення температури, при якому комп'ютер ще збоїтиме, але при значенні $a_{13,4}$ комп'ютер збоїти не буде.

З цього виходить, що в базу знань *KB* необхідно додати продукцію $p_{13,6}$ – «Якщо a_{13} (температура в приміщенні) = $a_{13,6}$ (температура_6), то перевірити a_2 ». При цьому продукції $p_{13,1}, p_{13,2}$ і $p_{13,3}$ можна замінити відношенням $r_{13,8}$.

База знань *KB* продукційною ЕС реального часу, яка здатна діагностувати будь-який стан системи *S*, може складатися також з наступних продукцій, що відображають знання експерта при проміжних міркуваннях:

– $p_{14,1}$ – «Якщо a_{14} (потік гарячого повітря) = $a_{14,1}$ (температура_1), то перевірити a_2 »;

– $p_{14,2}$ – «Якщо a_{14} (потік гарячого повітря) = $a_{14,2}$ (температура_2), то перевірити a_2 »;

– $p_{14,3}$ – «Якщо a_{14} (потік гарячого повітря) = $a_{14,3}$ (температура_3), то перевірити a_2 »;

– $p_{14,4}$ – «Якщо a_{14} (потік гарячого повітря) = $a_{14,4}$ (температура_4), то перевірити a_2 ».

Конкретні числові характеристики величин «температура_1» – «температура_4» залежать виключно від досвіду експерта, і різні експерти можуть задати різні значення цих температур.

У експерта також можна з'ясувати, які значення може приймати a_{14} . Нехай $a_{14} = \{a_{14,1}, a_{14,2}, a_{14,3}, a_{14,4}, a_{14,5}, a_{14,6}, a_{14,7}\}$, де $a_{14,i}$ – деяке значення температури. Потім експерт може визначити відношення $r_{14,8}$ характерності температури для збоїв комп'ютера: $a_{14,4} > a_{14,3} > a_{14,1} > a_{14,2} > a_{14,5} > a_{14,6} > a_{14,7}$. Нарешті, експерт може сказати, що $a_{14,5}$ – найменш характерне значення температури, при якій комп'ютер ще збоїтиме, але при значенні $a_{14,6}$ комп'ютер збоїти не буде.

З цього виходить, що в базу знань *KB* необхідно додати продукцію $p_{14,5}$ – «Якщо a_{14} (потік гарячого повітря) = $a_{14,5}$ (температура_5), то перевірити a_2 ». При цьому продукції $p_{14,1}$, $p_{14,2}$, $p_{14,3}$ і $p_{14,4}$ можна замінити відношенням $r_{14,8}$.

База знань *KB* продукційною ЕС реального часу, яка здатна діагностувати будь-який стан системи *S*, може складатися також з наступних продукцій, що відображають знання експерта при проміжних міркуваннях:

– $p_{15,1}$ – «Якщо a_{15} (вентилятор джерела живлення) = $a_{15,1}$ (швидкість_1), то перевірити a_2 »;

– $p_{15,2}$ – «Якщо a_{15} (вентилятор джерела живлення) = $a_{15,2}$ (швидкість_2), то перевірити a_2 »;

– $p_{15,3}$ – «Якщо a_{15} (вентилятор джерела живлення) = $a_{15,3}$ (швидкість_3), то

перевірити a_2 »;

– $p_{15,4}$ – «Якщо a_{15} (вентилятор джерела живлення) = $a_{15,4}$ (швидкість_4), то перевірити a_2 »;

– $p_{15,5}$ – «Якщо a_{15} (вентилятор джерела живлення) = $a_{15,5}$ (швидкість_5), то перевірити a_2 ».

Конкретні значення величин «швидкість_1» – «швидкість_5» залежать від досвіду експерта, і різні експерти можуть задати різні швидкості.

У експерта також можна з'ясувати, які значення може приймати параметр a_{15} . Нехай $a_{15} = \{a_{15,1}, a_{15,2}, a_{15,3}, a_{15,4}, a_{15,5}, a_{15,6}, a_{15,7}, a_{15,8}\}$, де $a_{15,i}$ – деяке значення швидкості обертання вентилятора. Потім експерт може визначити відношення $r_{15,8}$ характерності швидкості для збоїв комп'ютера: $a_{15,4} > a_{15,3} > a_{15,1} > a_{15,2} > a_{15,5} > a_{15,6} > a_{15,7} > a_{15,8}$. Нарешті, експерт може сказати, що $a_{15,7}$ – найменш характерне значення швидкості, при якій комп'ютер ще збоїтиме, але при значенні $a_{15,8}$ комп'ютер збоїти не буде.

З цього виходить, що в базу знань *KB* необхідно додати продукцію $p_{15,7}$ – «Якщо a_{15} (вентилятор джерела живлення) = $a_{15,7}$ (швидкість_7), то перевірити a_2 ». При цьому продукції $p_{15,1}, p_{15,2}, p_{15,3}, p_{15,4}, p_{15,5}$ можна замінити відношенням $r_{15,8}$.

База знань *KB* продукційною ЕС реального часу, яка здатна діагностувати будь-який стан системи S , може складатися також з наступних продукцій, що відображають знання експерта при проміжних міркуваннях:

– $p_{16,1}$ – «Якщо a_{16} (шар пилу) = $a_{16,1}$ (товщина_1), то перевірити a_2 »;

– $p_{16,2}$ – «Якщо a_{16} (шар пилу) = $a_{16,2}$ (товщина_2), то перевірити a_2 »;

– $p_{16,3}$ – «Якщо a_{16} (шар пилу) = $a_{16,3}$ (товщина_3), то перевірити a_2 »;

– $p_{16,4}$ – «Якщо a_{16} (шар пилу) = $a_{16,4}$ (товщина_4), то перевірити a_2 »;

– $p_{16,5}$ – «Якщо a_{16} (шар пилу) = $a_{16,5}$ (товщина_5), то перевірити a_2 »;

– $p_{16,6}$ – «Якщо a_{16} (шар пилу) = $a_{16,6}$ (товщина_6), то перевірити a_2 ».

Конкретні значення величин «товщина_1» – «товщина_6» залежать від досвіду експерта, і різні експерти можуть задати різні значення товщини.

У експерта також можна з'ясувати, які значення може приймати параметр a_{16} . Нехай $a_{16} = \{a_{16,1}, a_{16,2}, a_{16,3}, a_{16,4}, a_{16,5}, a_{16,6}, a_{16,7}, a_{16,8}, a_{16,9}\}$, де $a_{16,i}$ – деяке значення товщини шаруючи пилу усередині комп'ютера. Потім експерт може

визначити відношення $r_{16,8}$ характерності значень цієї товщини для збоїв комп'ютера: $a_{16,2} > a_{16,4} > a_{16,6} > a_{16,1} > a_{16,3} > a_{16,5} > a_{16,7} > a_{16,8} > a_{16,9}$. Нарешті, експерт може сказати, що $a_{16,8}$ – найменш характерне значення товщини, при якому комп'ютер ще збоїтиме, але при значенні $a_{16,9}$ комп'ютер збоїти не буде.

З цього виходить, що в базу знань *KB* необхідно додати продукцію $p_{16,8}$ – «Якщо a_{16} (шар пилу) = $a_{16,8}$ (товщина_8), то перевірити a_2 ». При цьому продукції $p_{16,1}, p_{16,2}, p_{16,3}, p_{16,4}, p_{16,5}, p_{16,6}$ можна замінити відношенням $r_{16,8}$.

База знань *KB* продукційною ЕС реального часу, яка здатна діагностувати будь-який стан системи S , може складатися також з наступних продукцій, що відображають знання експерта при проміжних міркуваннях:

– $p_{17,1}$ – «Якщо a_{17} (вентилятори, що охолоджують) = $a_{17,1}$ (швидкість_1), то перевірити a_2 »;

– $p_{17,2}$ – «Якщо a_{17} (вентилятори, що охолоджують) = $a_{17,2}$ (швидкість_2), то перевірити a_2 »;

– $p_{17,3}$ – «Якщо a_{17} (вентилятори, що охолоджують) = $a_{17,3}$ (швидкість_3), то перевірити a_2 ».

Конкретні значення величин «швидкість_1» – «швидкість_3» залежать від досвіду експерта, і різні експерти можуть задати різні швидкості.

У експерта також можна з'ясувати, які значення може приймати параметр a_{17} . Нехай $a_{17} = \{a_{17,1}, a_{17,2}, a_{17,3}, a_{17,4}, a_{17,5}\}$, де $a_{17,i}$ – деяка швидкість обертання вентиляторів, що охолоджують. Потім експерт може визначити відношення $r_{17,8}$ характерності цих швидкостей для збоїв комп'ютера: $a_{17,3} > a_{17,2} > a_{17,1} > a_{17,5} > a_{17,4}$. Нарешті, експерт може сказати, що $a_{17,5}$ – найменш характерна швидкість обертання, при якій комп'ютер ще збоїтиме, але при значенні $a_{17,4}$ комп'ютер збоїти не буде.

З цього виходить, що в базу знань *KB* необхідно додати продукцію $p_{17,5}$ – «Якщо a_{17} (вентилятори, що охолоджують) = $a_{17,5}$ (швидкість_5), то перевірити a_2 ». При цьому продукції $p_{17,1}, p_{17,2}$ і $p_{17,3}$ можна замінити відношенням $r_{17,8}$.

База знань *KB* продукційною ЕС реального часу може включати наступні продукції, такі, що відображають проміжні міркування експерта:

– $p_{18,1}$ – «Якщо a_{18} (температура радіаторів на платах) = $a_{18,1}$ (температура_1),

то перевірити a_2 »;

– $p_{18,2}$ – «Якщо a_{18} (температура радіаторів на платах) = $a_{18,2}$ (температура_2),

то перевірити a_2 »;

– $p_{18,3}$ – «Якщо a_{18} (температура радіаторів на платах) = $a_{18,3}$ (температура_3),

то перевірити a_2 »;

– $p_{18,4}$ – «Якщо a_{18} (температура радіаторів на платах) = $a_{18,4}$ (температура_4),

то перевірити a_2 »;

– $p_{18,5}$ – «Якщо a_{18} (температура радіаторів на платах) = $a_{18,5}$ (температура_5),

то перевірити a_2 ».

Конкретні значення величин «температура_1» – «температура_5» залежать від досвіду експерта, і різні експерти можуть задати різні об'єми.

У експерта також можна з'ясувати, які значення може приймати параметр a_{18} . Нехай $a_{18} = \{a_{18,1}, a_{18,2}, a_{18,3}, a_{18,4}, a_{18,5}, a_{18,6}, a_{18,7}\}$, де $a_{18,i}$ – деяка температура радіаторів на платах. Потім експерт може визначити відношення $r_{18,8}$ характерності температур для збоїв комп'ютера: $a_{18,1} > a_{18,2} > a_{18,3} > a_{18,5} > a_{18,4} > a_{18,6} > a_{18,7}$. Нарешті, експерт може сказати, що $a_{18,6}$ – найменш характерна температура, при якій комп'ютер ще збоїтиме, але при значенні $a_{18,7}$ комп'ютер збоїти не буде.

З цього виходить, що в базу знань *KB* необхідно додати продукцію $p_{18,6}$ – «Якщо a_{18} (температура радіаторів на платах) = $a_{18,6}$ (температура_6), то перевірити a_2 ». При цьому продукції $p_{18,1}$, $p_{18,2}$, $p_{18,3}$, $p_{18,4}$ і $p_{18,5}$ можна замінити відношенням $r_{18,8}$.

База знань *KB* продукційною ЕС реального часу може включати наступні продукції, такі, що відображають проміжні міркування експерта при постановці діагнозу:

– $p_{19,1}$ – «Якщо a_{19} (відстань між дисководами) = $a_{19,1}$ (відстань_1), то перевірити a_2 »;

– $p_{19,2}$ – «Якщо a_{19} (відстань між дисководами) = $a_{19,2}$ (відстань_2), то перевірити a_2 »;

– $p_{19,3}$ – «Якщо a_{19} (відстань між дисководами) = $a_{19,3}$ (відстань_3), то перевірити a_2 ».

Конкретні числові характеристики величин «відстань_1» – «відстань_3» залежать виключно від досвіду експерта, і різні експерти можуть задати різні значення цих відстаней.

У експерта також можна з'ясувати, які значення може приймати a_{19} . Нехай $a_{19} = \{a_{19,1}, a_{19,2}, a_{19,3}, a_{19,4}, a_{19,5}, a_{19,6}\}$, де $a_{19,i}$ – деяке значення відстані між дисководами. Потім експерт може визначити відношення $r_{19,8}$ характерності цієї відстані для збоїв комп'ютера: $a_{19,2} > a_{19,3} > a_{19,1} > a_{19,6} > a_{19,4} > a_{19,5}$. Нарешті, експерт може сказати, що $a_{19,6}$ – найменш характерна відстань, при якій комп'ютер ще збоїтиме, але при значенні $a_{19,4}$ комп'ютер збоїти не буде.

З цього виходить, що в базу знань *KB* необхідно додати продукцію $p_{19,6}$ – «Якщо a_{19} (відстань між дисководами) = $a_{19,6}$ (відстань_6), то перевірити a_2 ». Продукції $p_{19,1}, p_{19,2}, p_{19,3}$ можна замінити відношенням $r_{19,8}$.

База знань *KB* продукційною ЕС реального часу може включати наступні продукції, такі, що відображають проміжні міркування експерта:

– $p_{20,1}$ – «Якщо a_{20} (температура дисководів) = $a_{20,1}$ (температура_1), то перевірити a_2 »;

– $p_{20,2}$ – «Якщо a_{20} (температура дисководів) = $a_{20,2}$ (температура_2), то перевірити a_2 »;

– $p_{20,3}$ – «Якщо a_{20} (температура дисководів) = $a_{20,3}$ (температура_3), то перевірити a_2 »;

– $p_{20,4}$ – «Якщо a_{20} (температура дисководів) = $a_{20,4}$ (температура_4), то перевірити a_2 ».

Конкретні числові характеристики величин «температура_1» – «температура_4» залежать виключно від досвіду експерта, і різні експерти можуть задати різні значення цих температур.

У експерта також можна з'ясувати, які значення може приймати a_{20} . Нехай $a_{20} = \{a_{20,1}, a_{20,2}, a_{20,3}, a_{20,4}, a_{20,5}, a_{20,6}, a_{20,7}\}$, де $a_{20,i}$ – деяке значення температури. Потім експерт може визначити відношення $r_{20,8}$ характерності температури для збоїв комп'ютера: $a_{20,4} > a_{20,3} > a_{20,1} > a_{20,2} > a_{20,5} > a_{20,6} > a_{20,7}$. Нарешті, експерт може сказати, що $a_{20,5}$ – найменш характерне значення температури, при якій комп'ютер ще збоїтиме, але при значенні $a_{20,6}$ комп'ютер збоїти не буде.

З цього виходить, що в базу знань *KB* необхідно додати продукцію $p_{20,5}$ – «Якщо a_{20} (температура дисководів) = $a_{20,5}$ (температура_5), то перевірити a_2 ». При цьому продукції $p_{20,1}$, $p_{20,2}$, $p_{20,3}$ і $p_{20,4}$ можна замінити відношенням $r_{20,8}$.

База знань *KB* продукційною ЕС реального часу може включати наступні продукції, такі, що відображають проміжні міркування експерта при постановці діагнозу:

– $p_{21,1}$ – «Якщо a_{21} (програма для устаткування) = $a_{21,1}$ (програма_1), то перевірити a_3 »;

– $p_{21,2}$ – «Якщо a_{21} (програма для устаткування) = $a_{21,2}$ (програма_2), то перевірити a_3 »;

– $p_{21,3}$ – «Якщо a_{21} (програма для устаткування) = $a_{21,3}$ (програма_3), то перевірити a_3 »;

– $p_{21,4}$ – «Якщо a_{21} (програма для устаткування) = $a_{21,4}$ (програма_4), то перевірити a_3 »;

– $p_{21,5}$ – «Якщо a_{21} (програма для устаткування) = $a_{21,5}$ (програма_5), то перевірити a_3 ».

Конкретні значення «програма_1» – «програма_5» залежать від досвіду експерта, і різні експерти можуть задати різні програми.

У експерта також можна з'ясувати, які значення може приймати параметр a_{21} . Нехай $a_{21} = \{a_{21,1}, a_{21,2}, a_{21,3}, a_{21,4}, a_{21,5}, a_{21,6}, a_{21,7}, a_{21,8}\}$, де $a_{21,i}$ – деяка програма для устаткування. Потім експерт може визначити відношення $r_{21,8}$ характерності програм для збоїв комп'ютера: $a_{21,4} > a_{21,3} > a_{21,1} > a_{21,2} > a_{21,5} > a_{21,6} > a_{21,7} > a_{21,8}$. Нарешті, експерт може сказати, що $a_{21,7}$ – найменш характерна програма, при якій комп'ютер ще збоїтиме, але при значенні $a_{21,8}$ комп'ютер збоїти не буде.

З цього виходить, що в базу знань *KB* необхідно додати продукцію $p_{21,7}$ – «Якщо a_{21} (програма для устаткування) = $a_{21,7}$ (програма_7), то перевірити a_3 ». При цьому продукції $p_{21,1}$, $p_{21,2}$, $p_{21,3}$, $p_{21,4}$, $p_{21,5}$ можна замінити відношенням $r_{21,8}$.

База знань *KB* продукційною ЕС реального часу може включати наступні продукції, такі, що відображають проміжні міркування експерта при постановці діагнозу:

– $p_{22,1}$ – «Якщо a_{22} (версія драйвера) = $a_{22,1}$ (номер_1), то перевірити a_3 »;

- $p_{22,2}$ – «Якщо a_{22} (версія драйвера) = $a_{22,2}$ (номер_2), то перевірити a_3 »;
- $p_{22,3}$ – «Якщо a_{22} (версія драйвера) = $a_{22,3}$ (номер_3), то перевірити a_3 »;
- $p_{22,4}$ – «Якщо a_{22} (версія драйвера) = $a_{22,4}$ (номер_4), то перевірити a_3 »;
- $p_{22,5}$ – «Якщо a_{22} (версія драйвера) = $a_{22,5}$ (номер_5), то перевірити a_3 »;
- $p_{22,6}$ – «Якщо a_{22} (версія драйвера) = $a_{22,6}$ (номер_6), то перевірити a_3 ».

Конкретні значення «номер_1» – «номер_6» залежать від досвіду експерта, і різні експерти можуть задати різні значення номера.

У експерта також можна з'ясувати, які значення може приймати параметр a_{22} . Нехай $a_{22} = \{a_{22,1}, a_{22,2}, a_{22,3}, a_{22,4}, a_{22,5}, a_{22,6}, a_{22,7}, a_{22,8}, a_{22,9}\}$, де $a_{22,i}$ – деякий номер версії драйвера устаткування. Потім експерт може визначити відношення $r_{22,8}$ характерності номера версії драйвера для збоїв комп'ютера: $a_{22,2} > a_{22,4} > a_{22,6} > a_{22,1} > a_{22,3} > a_{22,5} > a_{22,7} > a_{22,8} > a_{22,9}$. Нарешті, експерт може сказати, що $a_{22,8}$ – найменш характерне значення версії, при якій комп'ютер ще збоїти, але при значенні $a_{22,9}$ комп'ютер збоїти не буде.

З цього виходить, що в базу знань *KB* необхідно додати продукцію $p_{22,8}$ – «Якщо a_{22} (версія драйвера) = $a_{22,8}$ (номер_8), то перевірити a_3 ». При цьому продукції $p_{22,1}, p_{22,2}, p_{22,3}, p_{22,4}, p_{22,5}, p_{22,6}$ можна замінити відношенням $r_{22,8}$.

База знань *KB* продукційною ЕС реального часу може включати наступні продукції, такі, що відображають проміжні міркування експерта при постановці діагнозу:

- $p_{23,1}$ – «Якщо a_{23} (гніздо плати розширення) = $a_{23,1}$ (гніздо_1), то перевірити a_3 »;
- $p_{23,2}$ – «Якщо a_{23} (гніздо плати розширення) = $a_{23,2}$ (гніздо_2), то перевірити a_3 »;
- $p_{23,3}$ – «Якщо a_{23} (гніздо плати розширення) = $a_{23,3}$ (гніздо_3), то перевірити a_3 ».

Конкретні значення «гніздо_1» – «гніздо_3» залежать від досвіду експерта, і різні експерти можуть задати різні гнізда.

У експерта також можна з'ясувати, які значення може приймати параметр a_{23} . Нехай $a_{23} = \{a_{23,1}, a_{23,2}, a_{23,3}, a_{23,4}, a_{23,5}\}$, де $a_{23,i}$ – деяке гніздо плати розширення. Потім експерт може визначити відношення $r_{23,8}$ характерності цих гнізд для збоїв

комп'ютера: $a_{23,3} > a_{23,2} > a_{23,1} > a_{23,5} > a_{23,4}$. Нарешті, експерт може сказати, що $a_{23,5}$ – найменш характерне гніздо, при якому комп'ютер ще збоїтиме, але при значенні $a_{23,4}$ комп'ютер збоїти не буде.

З цього виходить, що в базу знань *KB* необхідно додати продукцію $p_{23,5}$ – «Якщо a_{23} (гніздо плати розширення) = $a_{23,5}$ (гніздо_5), то перевірити a_3 ». При цьому продукції $p_{23,1}$, $p_{23,2}$ і $p_{23,3}$ можна замінити відношенням $r_{23,8}$.

База знань *KB* продукційною ЕС реального часу може включати наступні продукції, такі, що відображають проміжні міркування експерта:

- $p_{24,1}$ – «Якщо a_{24} (розділений порт) = $a_{24,1}$ (порт_1), то перевірити a_3 »;
- $p_{24,2}$ – «Якщо a_{24} (розділений порт) = $a_{24,2}$ (порт_2), то перевірити a_3 »;
- $p_{24,3}$ – «Якщо a_{24} (розділений порт) = $a_{24,3}$ (порт_3), то перевірити a_3 »;
- $p_{24,4}$ – «Якщо a_{24} (розділений порт) = $a_{24,4}$ (порт_4), то перевірити a_3 »;
- $p_{24,5}$ – «Якщо a_{24} (розділений порт) = $a_{24,5}$ (порт_5), то перевірити a_3 ».

Конкретні значення «порт_1» – «порт_5» залежать від досвіду експерта, і різні експерти можуть задати різні порти.

У експерта також можна з'ясувати, які значення може приймати параметр a_{24} . Нехай $a_{24} = \{a_{24,1}, a_{24,2}, a_{24,3}, a_{24,4}, a_{24,5}, a_{24,6}, a_{24,7}\}$, де $a_{24,i}$ – деякий порт. Потім експерт може визначити відношення $r_{24,8}$ характерності портів для збоїв комп'ютера: $a_{24,1} > a_{24,2} > a_{24,3} > a_{24,5} > a_{24,4} > a_{24,6} > a_{24,7}$. Нарешті, експерт може сказати, що $a_{24,6}$ – найменш характерний порт, при якому комп'ютер ще збоїтиме, але при значенні $a_{24,7}$ комп'ютер збоїти не буде.

З цього виходить, що в базу знань *KB* необхідно додати продукцію $p_{24,6}$ – «Якщо a_{24} (розділений порт) = $a_{24,6}$ (порт_6), то перевірити a_3 ». При цьому продукції $p_{24,1}$, $p_{24,2}$, $p_{24,3}$, $p_{24,4}$ і $p_{24,5}$ можна замінити відношенням $r_{24,8}$.

База знань *KB* продукційною ЕС реального часу може включати наступні продукції, такі, що відображають проміжні міркування експерта:

- $p_{25,1}$ – «Якщо a_{25} (драйвер графічної карти) = $a_{25,1}$ (версія_1), то перевірити a_4 »;
- $p_{25,2}$ – «Якщо a_{25} (драйвер графічної карти) = $a_{25,2}$ (версія_2), то перевірити a_4 »;
- $p_{25,3}$ – «Якщо a_{25} (драйвер графічної карти) = $a_{25,3}$ (версія_3), то перевірити a_4 ».

a_4 ».

Конкретні значення «версія_1» – «версія_3» залежать від досвіду експерта, і різні експерти можуть задати різні значення версій.

У експерта також можна з'ясувати, які значення може приймати a_{25} . Нехай $a_{25} = \{a_{25,1}, a_{25,2}, a_{25,3}, a_{25,4}, a_{25,5}, a_{25,6}\}$, де $a_{25,i}$ – деяка версія драйвера. Потім експерт може визначити відношення $r_{25,8}$ характерності версій для збоїв комп'ютера: $a_{25,2} > a_{25,3} > a_{25,1} > a_{25,6} > a_{25,4} > a_{25,5}$. Нарешті, експерт може сказати, що $a_{25,6}$ – найменш характерна версія, при якій комп'ютер ще збоїтиме, але при значенні $a_{25,4}$ комп'ютер збоїти не буде.

З цього виходить, що в базу знань *KB* необхідно додати продукцію $p_{25,6}$ – «Якщо a_{25} (драйвер графічної карти) = $a_{25,6}$ (версія_6), то перевірити a_4 ». Продукції $p_{25,1}, p_{25,2}, p_{25,3}$ можна замінити відношенням $r_{25,8}$.

База знань *KB* продукційною ЕС реального часу може включати наступні продукції, такі, що відображають проміжні міркування експерта:

- $p_{26,1}$ – «Якщо a_{26} (проблема при запуску старої програми) = $a_{26,1}$ (програма_1), то перевірити a_4 »;
- $p_{26,2}$ – «Якщо a_{26} (проблема при запуску старої програми) = $a_{26,2}$ (програма_2), то перевірити a_4 »;
- $p_{26,3}$ – «Якщо a_{26} (проблема при запуску старої програми) = $a_{26,3}$ (програма_3), то перевірити a_4 »;
- $p_{26,4}$ – «Якщо a_{26} (проблема при запуску старої програми) = $a_{26,4}$ (програма_4), то перевірити a_4 ».

Конкретні значення «програма_1» – «програма_4» залежать від досвіду експерта, і різні експерти можуть задати різні програми.

У експерта також можна з'ясувати, які значення може приймати a_{26} . Нехай $a_{26} = \{a_{26,1}, a_{26,2}, a_{26,3}, a_{26,4}, a_{26,5}, a_{26,6}, a_{26,7}\}$, де $a_{26,i}$ – деяка програма. Потім експерт може визначити відношення $r_{26,8}$ характерності програм для збоїв комп'ютера: $a_{26,4} > a_{26,3} > a_{26,1} > a_{26,2} > a_{26,5} > a_{26,6} > a_{26,7}$. Нарешті, експерт може сказати, що $a_{26,5}$ – найменш характерна програма, при якій комп'ютер ще збоїтиме, але при значенні $a_{26,6}$ комп'ютер збоїти не буде.

З цього виходить, що в базу знань *KB* необхідно додати продукцію $p_{26,5}$ –

«Якщо a_{26} (проблема при запуску старої програми) = $a_{26,5}$ (програма_5), то перевірити a_4 ». При цьому продукції $p_{26,1}$, $p_{26,2}$, $p_{26,3}$ і $p_{26,4}$ можна замінити відношенням $r_{26,8}$.

База знань *KB* продукційною ЕС реального часу може включати наступні продукції, такі, що відображають проміжні міркування експерта:

- $p_{27,1}$ – «Якщо a_{27} (проблема при одночасному запуску старої і нової програми) = $a_{27,1}$ (програми_1), то перевірити a_4 »;
- $p_{27,2}$ – «Якщо a_{27} (проблема при одночасному запуску старої і нової програми) = $a_{27,2}$ (програми_2), то перевірити a_4 »;
- $p_{27,3}$ – «Якщо a_{27} (проблема при одночасному запуску старої і нової програми) = $a_{27,3}$ (програми_3), то перевірити a_4 »;
- $p_{27,4}$ – «Якщо a_{27} (проблема при одночасному запуску старої і нової програми) = $a_{27,4}$ (програми_4), то перевірити a_4 »;
- $p_{27,5}$ – «Якщо a_{27} (проблема при одночасному запуску старої і нової програми) = $a_{27,5}$ (програми_5), то перевірити a_4 ».

Конкретні значення «програми_1» – «програми_5» залежать від досвіду експерта, і різні експерти можуть задати різні програми.

У експерта також можна з'ясувати, які значення може приймати параметр a_{27} . Нехай $a_{27} = \{a_{27,1}, a_{27,2}, a_{27,3}, a_{27,4}, a_{27,5}, a_{27,6}, a_{27,7}, a_{27,8}\}$, де $a_{27,i}$ – деякі конфліктуючі програми. Потім експерт може визначити відношення $r_{27,8}$ характерності програм для збоїв комп'ютера: $a_{27,4} > a_{27,3} > a_{27,1} > a_{27,2} > a_{27,5} > a_{27,6} > a_{27,7} > a_{27,8}$. Нарешті, експерт може сказати, що $a_{27,7}$ – найменш характерні програми, при яких комп'ютер ще збоїтиме, але при значенні $a_{27,8}$ комп'ютер збоїти не буде.

З цього виходить, що в базу знань *KB* необхідно додати продукцію $p_{27,7}$ – «Якщо a_{27} (проблема при одночасному запуску старої і нової програми) = $a_{27,7}$ (програми_7), то перевірити a_4 ». При цьому продукції $p_{27,1}$, $p_{27,2}$, $p_{27,3}$, $p_{27,4}$, $p_{27,5}$ можна замінити відношенням $r_{27,8}$.

База знань *KB* продукційною ЕС реального часу може включати наступні продукції, такі, що відображають проміжні міркування експерта:

- $p_{28,1}$ – «Якщо a_{28} (заміна файлів Windows) = $a_{28,1}$ (файли_1), то перевірити

a_4 »;

– $p_{28,2}$ – «Якщо a_{28} (заміна файлів Windows) = $a_{28,2}$ (файли_2), то перевірити

a_4 »;

– $p_{28,3}$ – «Якщо a_{28} (заміна файлів Windows) = $a_{28,3}$ (файли_3), то перевірити

a_4 »;

– $p_{28,4}$ – «Якщо a_{28} (заміна файлів Windows) = $a_{28,4}$ (файли_4), то перевірити

a_4 »;

– $p_{28,5}$ – «Якщо a_{28} (заміна файлів Windows) = $a_{28,5}$ (файли_5), то перевірити

a_4 »;

– $p_{28,6}$ – «Якщо a_{28} (заміна файлів Windows) = $a_{28,6}$ (файли_6), то перевірити

a_4 ».

Конкретні значення «файли_1» – «файли_6» залежать від досвіду експерта, і різні експерти можуть задати різні файли.

У експерта також можна з'ясувати, які значення може приймати параметр a_{28} . Нехай $a_{28} = \{a_{28,1}, a_{28,2}, a_{28,3}, a_{28,4}, a_{28,5}, a_{28,6}, a_{28,7}, a_{28,8}, a_{28,9}\}$, де $a_{28,i}$ – деякі файли Windows. Потім експерт може визначити відношення $r_{28,8}$ характерності файлів для збоїв комп'ютера: $a_{28,2} > a_{28,4} > a_{28,6} > a_{28,1} > a_{28,3} > a_{28,5} > a_{28,7} > a_{28,8} > a_{28,9}$. Нарешті, експерт може сказати, що $a_{28,8}$ – найменш характерні файли, при яких комп'ютер ще збоїтиме, але при значенні $a_{28,9}$ комп'ютер збоїти не буде.

З цього виходить, що в базу знань *KB* необхідно додати продукцію $p_{28,8}$ – «Якщо a_{28} (заміна файлів Windows) = $a_{28,8}$ (файли_8), то перевірити a_4 ». Продукції $p_{28,1}, p_{28,2}, p_{28,3}, p_{28,4}, p_{28,5}, p_{28,6}$ можна замінити відношенням $r_{28,8}$.

База знань *KB* продукційною ЕС реального часу може включати наступні продукції, такі, що відображають проміжні міркування експерта:

– $p_{29,1}$ – «Якщо a_{29} (розширена підтримка управління живленням) = $a_{29,1}$ (схема_1), то перевірити a_5 »;

– $p_{29,2}$ – «Якщо a_{29} (розширена підтримка управління живленням) = $a_{29,2}$ (схема_2), то перевірити a_5 »;

– $p_{29,3}$ – «Якщо a_{29} (розширена підтримка управління живленням) = $a_{29,3}$ (схема_3), то перевірити a_5 ».

Конкретні значення «схема_1» – «схема_3» залежать від досвіду експерта, і

різні експерти можуть задати різні схеми.

У експерта також можна з'ясувати, які значення може приймати параметр a_{29} . Нехай $a_{29} = \{a_{29,1}, a_{29,2}, a_{29,3}, a_{29,4}, a_{29,5}\}$, де $a_{29,i}$ – деяка схема розширеної підтримки управління живленням. Потім експерт може визначити відношення $r_{29,8}$ характерності цих схем для збоїв комп'ютера: $a_{29,3} > a_{29,2} > a_{29,1} > a_{29,5} > a_{29,4}$. Нарешті, експерт може сказати, що $a_{29,5}$ – найменш характерна схема, при якій комп'ютер ще збоїтиме, але при значенні $a_{29,4}$ комп'ютер збоїти не буде.

З цього виходить, що в базу знань *KB* необхідно додати продукцію $p_{29,5}$ – «Якщо a_{29} (розширена підтримка управління живленням) = $a_{29,5}$ (схема_5), то перевірити a_5 ». При цьому продукції $p_{29,1}$, $p_{29,2}$ і $p_{29,3}$ можна замінити відношенням $r_{29,8}$.

База знань *KB* продукційною ЕС реального часу може включати наступні продукції, такі, що відображають проміжні міркування експерта:

- $p_{30,1}$ – «Якщо a_{30} (заставка монітора) = $a_{30,1}$ (схема_1), то перевірити a_5 »;
- $p_{30,2}$ – «Якщо a_{30} (заставка монітора) = $a_{30,2}$ (схема_2), то перевірити a_5 »;
- $p_{30,3}$ – «Якщо a_{30} (заставка монітора) = $a_{30,3}$ (схема_3), то перевірити a_5 »;
- $p_{30,4}$ – «Якщо a_{30} (заставка монітора) = $a_{30,4}$ (схема_4), то перевірити a_5 »;
- $p_{30,5}$ – «Якщо a_{30} (заставка монітора) = $a_{30,5}$ (схема_5), то перевірити a_5 ».

Конкретні значення «схема_1» – «схема_5» залежать від досвіду експерта, і різні експерти можуть задати різні схеми.

У експерта також можна з'ясувати, які значення може приймати параметр a_{30} . Нехай $a_{30} = \{a_{30,1}, a_{30,2}, a_{30,3}, a_{30,4}, a_{30,5}, a_{30,6}, a_{30,7}\}$, де $a_{30,i}$ – деяка схема управління екранною заставкою. Потім експерт може визначити відношення $r_{30,8}$ характерності схем для збоїв комп'ютера: $a_{30,1} > a_{30,2} > a_{30,3} > a_{30,5} > a_{30,4} > a_{30,6} > a_{30,7}$. Нарешті, експерт може сказати, що $a_{30,6}$ – найменш характерна схема, при якій комп'ютер ще збоїтиме, але при значенні $a_{30,7}$ комп'ютер збоїти не буде.

З цього виходить, що в базу знань *KB* необхідно додати продукцію $p_{30,6}$ – «Якщо a_{30} (заставка монітора) = $a_{30,6}$ (схема_6), то перевірити a_5 ». При цьому продукції $p_{30,1}$, $p_{30,2}$, $p_{30,3}$, $p_{30,4}$ і $p_{30,5}$ можна замінити відношенням $r_{30,8}$.

База знань *KB* продукційною ЕС реального часу може включати наступні продукції, такі, що відображають проміжні міркування експерта:

– $p_{31,1}$ – «Якщо a_{31} (управління живленням монітора) = $a_{31,1}$ (схема_1), то перевірити a_5 »;

– $p_{31,2}$ – «Якщо a_{31} (управління живленням монітора) = $a_{31,2}$ (схема_2), то перевірити a_5 »;

– $p_{31,3}$ – «Якщо a_{31} (управління живленням монітора) = $a_{31,3}$ (схема_3), то перевірити a_5 ».

Конкретні значення «схема_1» – «схема_3» залежать від досвіду експерта, і різні експерти можуть задати різні схеми.

У експерта також можна з'ясувати, які значення може приймати a_{31} . Нехай $a_{31} = \{a_{31,1}, a_{31,2}, a_{31,3}, a_{31,4}, a_{31,5}, a_{31,6}\}$, де $a_{31,i}$ – деяка схема управління живленням монітора. Потім експерт може визначити відношення $r_{31,8}$ характерності схем для збоїв комп'ютера: $a_{31,2} > a_{31,3} > a_{31,1} > a_{31,6} > a_{31,4} > a_{31,5}$. Нарешті, експерт може сказати, що $a_{31,6}$ – найменш характерна схема, при якій комп'ютер ще збоїтиме, але при значенні $a_{31,4}$ комп'ютер збоїти не буде.

З цього виходить, що в базу знань *KB* необхідно додати продукцію $p_{31,6}$ – «Якщо a_{31} (управління живленням монітора) = $a_{31,6}$ (схема_6), то перевірити a_5 ». Продукції $p_{31,1}, p_{31,2}, p_{31,3}$ можна замінити відношенням $r_{31,8}$.

База знань *KB* продукційною ЕС реального часу може включати наступні продукції, такі, що відображають проміжні міркування експерта:

– $p_{32,1}$ – «Якщо a_{32} (працює програм) = $a_{32,1}$ (число_1), то перевірити a_6 »;

– $p_{32,2}$ – «Якщо a_{32} (працює програм) = $a_{32,2}$ (число_2), то перевірити a_6 »;

– $p_{32,3}$ – «Якщо a_{32} (працює програм) = $a_{32,3}$ (число_3), то перевірити a_6 »;

– $p_{32,4}$ – «Якщо a_{32} (працює програм) = $a_{32,4}$ (число_4), то перевірити a_6 ».

Конкретні значення «число_1» – «число_4» залежать від досвіду експерта, і різні експерти можуть задати різні числа.

У експерта також можна з'ясувати, які значення може приймати a_{32} . Нехай $a_{32} = \{a_{32,1}, a_{32,2}, a_{32,3}, a_{32,4}, a_{32,5}, a_{32,6}, a_{32,7}\}$, де $a_{32,i}$ – деяке число. Потім експерт може визначити відношення $r_{32,8}$ характерності числа працюючих програм для збоїв комп'ютера: $a_{32,4} > a_{32,3} > a_{32,1} > a_{32,2} > a_{32,5} > a_{32,6} > a_{32,7}$. Нарешті, експерт може сказати, що $a_{32,5}$ – найменш характерне число працюючих програм, при якому комп'ютер ще збоїтиме, але при значенні $a_{32,6}$ комп'ютер збоїти не буде.

З цього виходить, що в базу знань *KB* необхідно додати продукцію $p_{32,5}$ – «Якщо a_{32} (працює програм) = $a_{32,5}$ (число_5), то перевірити a_6 ». При цьому продукції $p_{32,1}$, $p_{32,2}$, $p_{32,3}$ і $p_{32,4}$ можна замінити відношенням $r_{32,8}$.

База знань *KB* продукційною ЕС реального часу може включати наступні продукції, такі, що відображають проміжні міркування експерта:

- $p_{33,1}$ – «Якщо a_{33} (пам'ять зайнята) = $a_{33,1}$ (програма_1), то перевірити a_6 »;
- $p_{33,2}$ – «Якщо a_{33} (пам'ять зайнята) = $a_{33,2}$ (програма_2), то перевірити a_6 »;
- $p_{33,3}$ – «Якщо a_{33} (пам'ять зайнята) = $a_{33,3}$ (програма_3), то перевірити a_6 »;
- $p_{33,4}$ – «Якщо a_{33} (пам'ять зайнята) = $a_{33,4}$ (програма_4), то перевірити a_6 »;
- $p_{33,5}$ – «Якщо a_{33} (пам'ять зайнята) = $a_{33,5}$ (програма_5), то перевірити a_6 ».

Конкретні значення «програма_1» – «програма_5» залежать від досвіду експерта, і різні експерти можуть задати різні програми.

У експерта також можна з'ясувати, які значення може приймати параметр a_{33} . Нехай $a_{33} = \{a_{33,1}, a_{33,2}, a_{33,3}, a_{33,4}, a_{33,5}, a_{33,6}, a_{33,7}, a_{33,8}\}$, де $a_{33,i}$ – деякі програми, що не звільняють системні ресурси після закінчення роботи. Потім експерт може визначити відношення $r_{33,8}$ характерності таких програм для збоїв комп'ютера: $a_{33,4} > a_{33,3} > a_{33,1} > a_{33,2} > a_{33,5} > a_{33,6} > a_{33,7} > a_{33,8}$. Нарешті, експерт може сказати, що $a_{33,7}$ – найменш характерні програми, при яких комп'ютер ще збоїтиме, але при значенні $a_{33,8}$ комп'ютер збоїти не буде.

З цього виходить, що в базу знань *KB* необхідно додати продукцію $p_{33,7}$ – «Якщо a_{33} (пам'ять зайнята) = $a_{33,7}$ (програма_7), то перевірити a_6 ». При цьому продукції $p_{33,1}$, $p_{33,2}$, $p_{33,3}$, $p_{33,4}$, $p_{33,5}$ можна замінити відношенням $r_{33,8}$.

База знань *KB* продукційною ЕС реального часу може включати наступні продукції, такі, що відображають проміжні міркування експерта:

- $p_{34,1}$ – «Якщо a_{34} (проводка) = $a_{34,1}$ (схема_1), то перевірити a_7 »;
- $p_{34,2}$ – «Якщо a_{34} (проводка) = $a_{34,2}$ (схема_2), то перевірити a_7 »;
- $p_{34,3}$ – «Якщо a_{34} (проводка) = $a_{34,3}$ (схема_3), то перевірити a_7 »;
- $p_{34,4}$ – «Якщо a_{34} (проводка) = $a_{34,4}$ (схема_4), то перевірити a_7 »;
- $p_{34,5}$ – «Якщо a_{34} (проводка) = $a_{34,5}$ (схема_5), то перевірити a_7 »;
- $p_{34,6}$ – «Якщо a_{34} (проводка) = $a_{34,6}$ (схема_6), то перевірити a_7 ».

Конкретні значення «схема_1» – «схема_6» залежать від досвіду експерта, і

різні експерти можуть задати різні схеми.

У експерта також можна з'ясувати, які значення може приймати параметр a_{34} . Нехай $a_{34} = \{a_{34,1}, a_{34,2}, a_{34,3}, a_{34,4}, a_{34,5}, a_{34,6}, a_{34,7}, a_{34,8}, a_{34,9}\}$, де $a_{34,i}$ – деякі схеми проводки. Потім експерт може визначити відношення $r_{34,8}$ характерності схем проводки для збоїв комп'ютера: $a_{34,2} > a_{34,4} > a_{34,6} > a_{34,1} > a_{34,3} > a_{34,5} > a_{34,7} > a_{34,8} > a_{34,9}$. Нарешті, експерт може сказати, що $a_{34,8}$ – найменш характерна схема, при якій комп'ютер ще збоїтиме, але при значенні $a_{34,9}$ комп'ютер збоїти не буде.

З цього виходить, що в базу знань *KB* необхідно додати продукцію $p_{34,8}$ – «Якщо a_{34} (проводка) = $a_{34,8}$ (схема_8), то перевірити a_7 ». Продукції $p_{34,1}, p_{34,2}, p_{34,3}, p_{34,4}, p_{34,5}, p_{34,6}$ можна замінити відношенням $r_{34,8}$.

База знань *KB* продукційною ЕС реального часу може включати наступні продукції, такі, що відображають проміжні міркування експерта:

- $p_{35,1}$ – «Якщо a_{35} (напруга) = $a_{35,1}$ (зниження_1), то перевірити a_7 »;
- $p_{35,2}$ – «Якщо a_{35} (напруга) = $a_{35,2}$ (зниження_2), то перевірити a_7 »;
- $p_{35,3}$ – «Якщо a_{35} (напруга) = $a_{35,3}$ (зниження_3), то перевірити a_7 ».

Конкретні значення «зниження_1» – «зниження_3» залежать від досвіду експерта, і різні експерти можуть задати різні зниження.

У експерта також можна з'ясувати, які значення може приймати параметр a_{35} . Нехай $a_{35} = \{a_{35,1}, a_{35,2}, a_{35,3}, a_{35,4}, a_{35,5}\}$, де $a_{35,i}$ – деяке зниження напруги в джерелі живлення. Потім експерт може визначити відношення $r_{35,8}$ характерності цих знижень для збоїв комп'ютера: $a_{35,3} > a_{35,2} > a_{35,1} > a_{35,5} > a_{35,4}$. Нарешті, експерт може сказати, що $a_{35,5}$ – найменш характерне зниження, при якому комп'ютер ще збоїтиме, але при значенні $a_{35,4}$ комп'ютер збоїти не буде.

З цього виходить, що в базу знань *KB* необхідно додати продукцію $p_{35,5}$ – «Якщо a_{35} (напруга) = $a_{35,5}$ (зниження_5), то перевірити a_7 ». При цьому продукції $p_{35,1}, p_{35,2}$ і $p_{35,3}$ можна замінити відношенням $r_{35,8}$.

База знань *KB* продукційною ЕС реального часу може включати наступні продукції, такі, що відображають проміжні міркування експерта при постановці діагнозу:

- $p_{36,1}$ – «Якщо a_{36} (відсутність живлення) = $a_{36,1}$ (час_1), то перевірити a_7 »;
- $p_{36,2}$ – «Якщо a_{36} (відсутність живлення) = $a_{36,2}$ (час_2), то перевірити a_7 »;

- $p_{36,3}$ – «Якщо a_{36} (відсутність живлення) = $a_{36,3}$ (час_3), то перевірити a_7 »;
- $p_{36,4}$ – «Якщо a_{36} (відсутність живлення) = $a_{36,4}$ (час_4), то перевірити a_7 »;
- $p_{36,5}$ – «Якщо a_{36} (відсутність живлення) = $a_{36,5}$ (час_5), то перевірити a_7 ».

Конкретні значення «час_1» – «час_5» залежать від досвіду експерта, і різні експерти можуть задати різний час.

У експерта також можна з'ясувати, які значення може приймати параметр a_{36} . Нехай $a_{36} = \{a_{36,1}, a_{36,2}, a_{36,3}, a_{36,4}, a_{36,5}, a_{36,6}, a_{36,7}\}$, де $a_{36,i}$ – якийсь час відключення живлення. Потім експерт може визначити відношення $r_{36,8}$ характерності часу відключення живлення для збоїв комп'ютера: $a_{36,1} > a_{36,2} > a_{36,3} > a_{36,5} > a_{36,4} > a_{36,6} > a_{36,7}$. Нарешті, експерт може сказати, що $a_{36,6}$ – найменш характерний час, при якому комп'ютер ще збоїтиме, але при значенні $a_{36,7}$ комп'ютер збоїти не буде.

З цього виходить, що в базу знань *KB* необхідно додати продукцію $p_{36,6}$ – «Якщо a_{36} (відсутність живлення) = $a_{36,6}$ (час_6), то перевірити a_7 ». При цьому продукції $p_{36,1}, p_{36,2}, p_{36,3}, p_{36,4}$ і $p_{36,5}$ можна замінити відношенням $r_{36,8}$.

3.2 Синтез нейронної мережі в програмовану логіку

В першому розділі дипломної роботи наголошувалося, що інтеграція нейронних мереж з продукційними правилами – важливий напрям розвитку систем штучного інтелекту.

Апаратна реалізація нейронних мереж (НМ) забезпечує максимально можливу швидкодію при вирішенні специфічних нейромережових задач, таких як діагностика, розпізнавання образів, управління, прогнозування і т. д. Такі НМ дозволяють реалізувати переваги властивого їм паралелізму і працюють на декілька порядків швидше в порівнянні з їх програмною симуляцією. Особливо це актуально для задач реального часу, пов'язаних, наприклад, з діагностикою комп'ютерних систем із застосуванням ЕС, в яких знання представлені у вигляді продукцій.

Використання сучасних перепрограмованих користувачем базових матричних кристалів (FPGA) для реалізації НМ можна вважати альтернативою по

відношенню до тільки програмних або тільки апаратних варіантів імплементації. Сучасні FPGA мають значний об'єм ресурсів – до 10 млн. системних елементів на кристал, високу продуктивність з робочими системними частотами до 420 МГц і можливістю реконфігурації кристала безпосередньо на робочому місці [52].

Застосування сучасної широкої номенклатури FPGA, а також засобів їх автоматизованого проектування і відладки, дозволять об'єднати переваги програмного і апаратного способів реалізації НМ. При цьому можуть бути вирішені питання, пов'язані з проблемою зменшення вартості і термінів розробки, а також зниження апаратних витрат при схемній реалізації НМ і подальшій її модифікації.

Оскільки в даному підрозділі основна увага приділяється методичним питанням синтезу НМ в програмовану логіку, то тут допускаються наступні спрощення: структура НМ заздалегідь задана, оптимізована і навчена по алгоритму зворотного розповсюдження; відомі кількість нейронних елементів з ваговими коефіцієнтами кожного з нейронів, необхідних для правильної роботи НМ.

В якості прикладу для реалізації розглянемо НМ, приведену в [53], де описана цільова задача медичної діагностики, моделювання і оптимізацію структури заданої НМ. Не дивлячись на те, що статистична медична діагностика не відноситься до за задач реального часу, методика синтезу НМ, описана нижче, застосовна для будь-якої довільно заданої структури НМ.

У даному підрозділі міститься приклад реалізації НМ і опис етапів її розробки від функціонального опису моделі до оцінки тимчасових і апаратних характеристик. Приведена оцінка оптимальності використання моделі НМ, отриманої після опису використовуваного нейроалгоритму в коді VHDL.

Для реалізації НМ в апаратному вигляді отримана структурна модель НМ представлена в кодах мови VHDL. Отримана на мові VHDL модель містить функціональну модель НМ і складений «тестбенч» з необхідними тестами для перевірки функціонування до синтезу, після синтезу і після розміщення логіки на чіпі. Наступним етапом модель синтезована в цифрову логіку для подальшої імплементації на таких цифрових пристроях як FPGA.

VHDL (VHSIC Hardware Description Language, IEEE Standard 1076-1993) [54] є мовою опису апаратних засобів загального призначення, використовується для опису і моделювання функціонування широкого ряду цифрових систем і дозволяє, з ухваленням відповідного із стилів кодування, синтез кодів в цифрову логіку [55]. Мова дозволяє проектувальникові використовувати однаковий опис моделі при створенні і симуляції попередньої концепції проекту і, якщо вона була завершена успішно, використовувати цю модель для її синтезу в цифрову логіку. При використанні методології проектування «зверху-вниз» опис проектного пристрою спочатку формується на системному/алгоритмічному рівні, і після цього модель аналізується на функціональну правильність. Потім генеруються детальніші моделі із збільшенням точності опису і обліком всіх аспектів апаратної реалізації.

Для реалізації даної НМ була вибрана плата Xilinx Virtex II FPGA. Вона має вбудовані блоки помножувачів 18x18, спеціальну логіку прискореного перенесення для високошвидкісних арифметичних операцій, часткове реконфігурування, гнучкі логічні ресурси: до 122 880 тригерів, до 122 880 16-ти розрядних регістрів зсуву на базі LUT, підтримка багатовходових помножувачів і логічних функцій. Зроблений за технологією 0.15-мкм з 8-шаровою металізацією і 0.12-мкм швидкодіючими транзисторами. Часткове реконфігурування дозволяє проводити дуже швидкі наростаючі зміни властиві алгоритму навчання. Архітектура, що базується на мультиплексорах, гарантує відсутність конфліктних ситуацій при проходженні сигналів, і таким чином, що довільні конфігураційні дані не приведуть плату до самознищення.

В процесі синтезу основне завдання – це опис моделі НМ стилем, що синтезується. Якщо код VHDL компілюється і правильно моделюється, то не обов'язково, що він правильно синтезуватиметься. І навіть якщо код VHDL правильно синтезується, результуюча реалізація може бути неефективною. Зазвичай, інструментальні засоби синтезу в якості входу прийматимуть тільки підмножину VHDL. Інші зміни повинні бути зроблені в коді VHDL, тому інструмент синтезу "розуміє" наміри проектувальника. Щоб провести ефективну реалізацію, можуть бути потрібними подальші зміни в коді VHDL.

Наприклад, коли використовуються цілочисельні сигнали, важливо визначити цілочисельний діапазон. Якщо такий діапазон не визначений, VHDL синтезатор може інтерпретувати цілочисельний сигнал для представлення 32-розрядного регістра, оскільки максимальний розмір цілого числа VHDL – 32 біта. Коли цілочисельний діапазон визначений, більшість синтезаторів реалізує цілочисельне додавання і віднімання, використовуючи двійкові суматори з відповідним числом бітів. Взагалі, коли сигналу VHDL присвоюється значення, він зберігатиме це значення до тих пір, поки йому не присвоять нове значення.

Кожен синтезатор має власний функціональний пакет для операцій, які зазвичай використовуються в моделях апаратури. IEEE надає стандартний синтезуючий пакет, що включає функції для арифметичних операцій над `bit_vectors` і `std_logic` векторами. Пакет `numeric_bit` визначає арифметичні операції `bit_vector` векторів. Пакет містить два класи масивів для подачі беззнакових і знакових бітових значень:

```
type unsigned is array (natural range <>) of bit;  
type signed is array (natural range <>) of bit;
```

Числа із знаками представляються в двійковому коді. Пакет містить версії арифметичних операцій, операцій відношення, логічних і операцій зсуву, які перезавантажуються, так само як і функції перетворення. Пакет `numeric_std` визначає аналогічні операції для `std_logic` векторів. Типи без знаку і із знаком визначаються як масиви `std_logic` векторів замість масивів бітів. Тип `UNSIGNED` заснований на IEEE-стандартній логіці, також замінені вся `bit` логіка на тип `STD_LOGIC`.

VHDL-код проекту містить функції, яких немає в синтезуючих бібліотеках. З цієї причини замінені `std_logic` на тип `UNSIGNED` або `signed`, щоб можна було використовувати визначених для цих типів операторів “+” і “*”, що перезавантажуються.

У програмі використані пакети бібліотеки IEEE:

```
IEEE.std_logic_1164.all;  
ieee.numeric_std.all;  
ieee.std_logic_unsigned.all;  
ieee.std_logic_signed.all;
```

Також створені власні пакети (наприклад, пакет `global_matrixs` містить опис матриці вагів першого нейрона першого шару), що містять опис типу сигналів вагів синапсів нейрона. Надалі можна користуватися цими пакетами, ввівши наступну стрічку в розділі опису бібліотек, що використовуються:

```
use global_matrixs.all;
```

Основні проблеми, які можуть виникнути при реалізації НМ в реконфігурованій апаратурі, пов'язані з представленням чисел з плаваючою крапкою і нелінійної активаційної функції нейрона.

Робота з числами з плаваючою крапкою є проблемою, оскільки арифметичні операції і схеми для чисел з плаваючою крапкою набагато складніші, ніж для цілих чисел, повільніше працюють і займають велику площу кристалу. Відповідним рішенням для удосконалення продуктивності проекту і зменшення апаратних витрат є конвертація чисел з плаваючою крапкою в числа типу `integer`. Звичайно, це накладає деякі обмеження на точність, але в даному випадку були досягнуті добрі результати. Діапазон зміни вагів $(-1;1)$ – число з плаваючою крапкою – описується на мові VHDL як число у форматі `real`. Числа формату `real` не синтезуються, тому необхідно перейти до представлення вагів в цілочисельному форматі. Для цього зроблене наступне. Описаний сигнал, що представляє ваговий коефіцієнт синапсу нейрона, числом типу `unsigned` (ціле без знаку), маючи на увазі, що таким чином створюється двійкове представлення дробової частини вагового коефіцієнта. Тобто, якщо ваговий коефіцієнт 2-го синапсу першого нейрона $W_{1,2} = 0.5$ в двійковій системі числення представляється як `0.1`, то в даному проекті він представлений як `1000 0000`.

У програмі, яка реалізовує НМ, на вхідні сигнали X_i подаються ознаки у вже

нормалізованому вигляді. Тобто сигнали X_i теж знаходяться в діапазоні $(-1;1)$ і представлені типом `unsigned`.

Результат множення $X*W$ буде менше одиниці. У моделі нейрона це синапс, який повинен бути підсумовуваний з іншими синапсами нейрона. Для того, щоб результат множення представити в зручному вигляді і згодом використовувати в підсумовуванні, множник Q_{pi} зсувається на 8 розрядів вправо, а потім беруться молодші 8 бітів. Таким чином, отриманий нами множник округляється до 8-ми бітів:

```
Qout_frac1 := Qp3 Srl n;  
Q_frac1 := Qout_frac1(n-1 downto 0);
```

Інша проблема представлення арифметичних операцій в цифровій апаратурі полягає в реалізації активаційної функції нейрона. Деякі активаційні функції, наприклад такі як сигмоїдна (використовується в моделі нейрона при створенні моделі за допомогою пакету NeuroPro 0.5), потребують деяких модифікацій для того, щоб спростити її для проектування в апаратурі. У нашому випадку, сигмоїдна функція заміщена кусочно-лінійною функцією (рисунок 3.1).

Таким чином, активаційна функція описана на мові VHDL таким чином:

```
if ADD_frac2 <= COMP then  
    Q_B:="11111111";  
else Q1:= ADDITION*ADD_f;  
    Q1:=Q1 sll 3;  
    Q_B:= Q1(10 downto 3);  
end if;
```

Тобто, якщо значення на виході суматора лежить в діапазоні від $-0,2$ до $0,2$, то Y обчислюватиметься по функції $y=5x$, в іншому випадку функція буде рівна -1 або 1 ($1 - y$ в випадку, якщо вихід суматора позитивний).

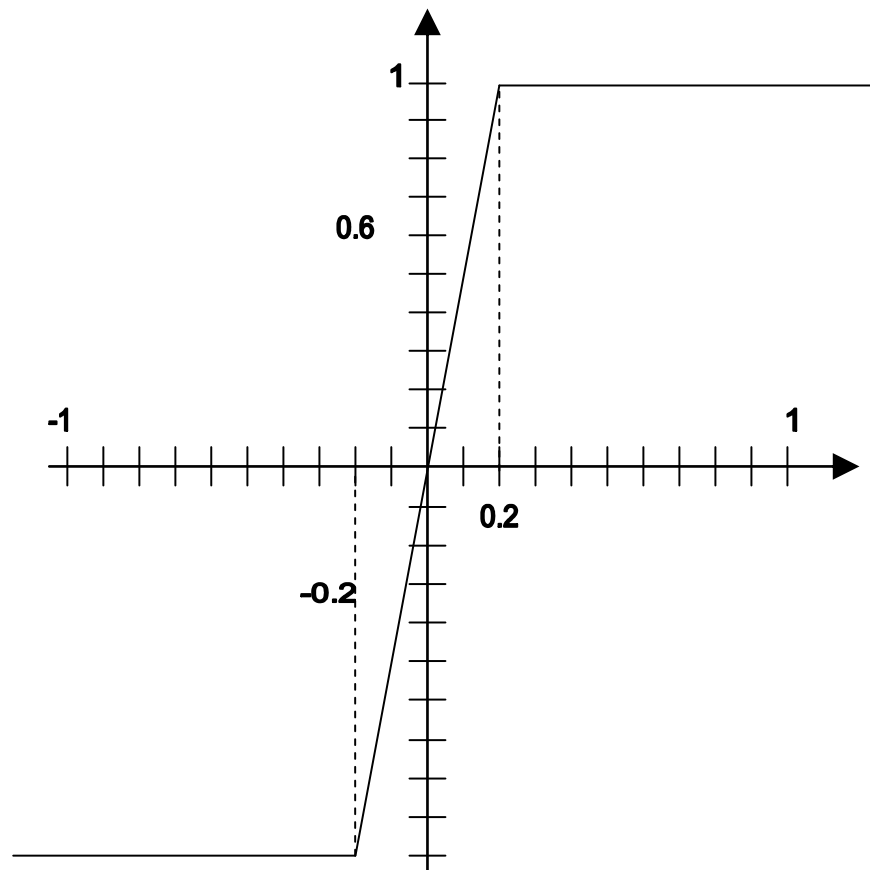


Рисунок 3.1 – Активаційна функція нейрона

Отримана НМ складається з трьох шарів. Вхідний шар має чотири нейрони, на яких паралельно поступає інформація з тестової вибірки. Кожен нейрон має різну кількість синапсів – вхідні сигнали (ознаки) поступають на кожен з нейронів вхідного шару. Перший нейрон має три синапси, другий – 14, третій – 12, четвертий – 13.

Другий прихований шар складається з трьох нейронів, які обробляють дані, що поступили від нейронів першого шару. Кожен з нейронів другого шару має 4 вхідних шини, що представляє синапси нейрона і сигнал bias – сигнал зсуву.

Третій шар – шар підсумовування складається з двох елементів, що підсумовують, які підсумовують виходи від 3-х нейронів попереднього шару.

Матриця вагів описується як одновимірний масив, елементами якого є вагові коефіцієнти кожного з синапсів нейрона. Матриця вагів окрема для кожного з нейронів НМ.

При описі моделі НМ на мові VHDL були створені наступні компоненти:

чотири нейрони першого шару, що містять опис в чотирьох окремих файлах, – `sample3.vhdl`, `neuron11_2.vhdl`, `neuron11_3.vhdl`, `neuron11_4.vhdl`; один компонент, що описує нейрони другого шару (вони однакові по архітектурі) у файлі `neuron21_comp.vhdl`, і елемент шару, що підсумовує у файлі `sum_component.vhdl`. Другий шар складається з трьох компонентів нейрона другого шару. Шар, що підсумовує, складається з двох елементів і схеми порівняння, оскільки на вихідний тригер поступає результат з того елемента, що підсумовує, чий вихід більший.

Кожен з компонентів, що представляють нейрон, містить в собі певну кількість синапсів, один суматор і порогову функцію. Шина, що відповідає за подачу вхідних сигналів, а також інші вхідні шини мають розмірність 8 бітів (7 downto 0), оскільки велика точність не потрібна.

Синапс в моделі нейрона реалізований операцією множення $X_{i,j} * W_{i,j}$, де $X_{i,j}$ – вхідна ознака, що поступає на j -ий синапс i -го нейрону; $W_{i,j}$ – ваговий коефіцієнт j -го синапсу i -го нейрону.

Підсумовування оперує з числами у форматі `signed`. Для цього, перш ніж підсумовувати, необхідно перетворити результат множення з формату `unsigned` в `signed` з врахуванням знаку множника.

В ході роботи над проектом з'ясовано, що синтезована схема повністю залежить від способу опису функціонування пристрою.

Опис НМ і контролера зроблений на структурному рівні за допомогою мови VHDL, і в результаті, після синтезу отримані такі елементи логіки як регістри, суматори, тригери і інші логічні блоки, які зіставлені з відповідними об'єктам структури.

3.3 Опис синтезованої моделі нейронної мережі

Для функціонального моделювання використаний пакет фірми Aldec ACTIVE-HDL 5.0.

Результати симуляції схеми до синтезу і після нього співпадають, що означає правильність синтезу схеми.

Синтез файлів VHDL здійснений з використанням пакету SynplifyPro 7.0 від Synplcity® і орієнтований на VIRTEX-II сімейство ПЛІС з архітектурою FPGA. Використана мікросхема XC2V6000 сімейства VIRTEX-II. Вона містить 6 млн. системних елементів, 144 блоки помножувачів і блокову пам'ять ємністю 2592 Кбіт.

Synplify Pro – пакет, призначений для синтезу логічних елементів, орієнтований на FPGA (Field Programmable Gate Arrays) і Cpld, розроблений фірмою Synplcity®. Робота з Synplify Pro починається з високорівневих проектів, написаних на мовах Verilog і VHDL (мови опису апаратури). Використовуючи власну Behavior Extracting Synthesis Technology (B.E.S.T.) технологію, пакет конвертує HDL в маленькі, високопродуктивні списки з'єднань проекту (design netlists), оптимізовані для загальнопоширених постачальників технологій. При необхідності, Synplify Pro може проводити VHDL і Verilog списки з'єднань після синтезу (post-synthesis netlists), які можна використовувати для моделювання після синтезу, щоб перевірити правильність синтезу.

Розроблений пристрій складається з контролера і самої НМ. Блок контролю працює на системній частоті 20 МГц і надає необхідні сигнали контролю. Контролер керує роботою НМ. Він синхронізує роботу кожного шару і повідомляє, коли можна подавати на входи наступний тестовий вектор.

Схема НМ синтезована за допомогою SynplifyPro 7.0. З рисунку 3.2 видно чотири нейрони у вхідному шарі і два компоненти прихованого шару і шару підсумовування.

Кожен з нейронів містить певну кількість логічних елементів, що реалізують функцію нейронного елемента.

На рисунку 3.3 зображена схема першого нейрона вхідного шару. Він складається з 5-ти регістрів (для інвертування шини), 2-х суматорів, 4-х DE-тригерів, 10-ти мультиплексорів 2-в-1, 4-х помножувачів, 1 тригер-латч, 2-х входових елементів «або», «і» і що «виключаючого або».

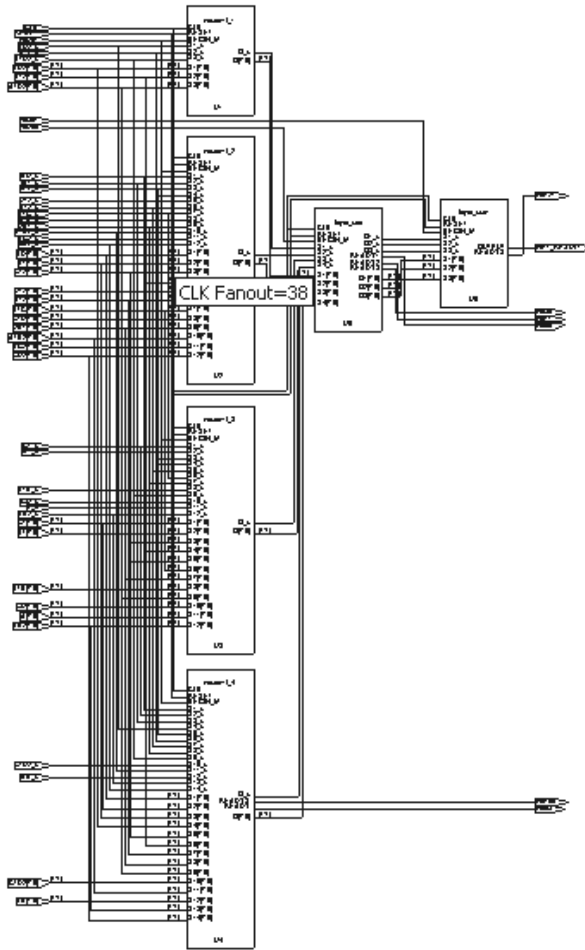


Рисунок 3.2 – Схема нейронної мережі

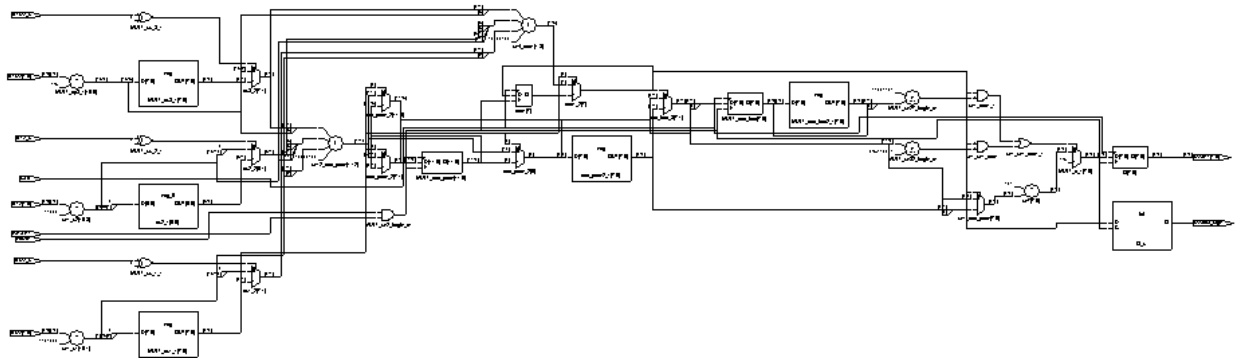


Рисунок 3.3 – Схема першого нейрона вхідного шару

Необхідно перевірити правильність функціонування пристрою після синтезу. Це моделювання покаже, що синтез пройшов успішно. Симуляція після етапу place and route необхідна для впевненості, що проект працюватиме з необхідною швидкістю, беручи до уваги час завантаження логічних елементів і затримку розповсюдження сигналу в лініях.

У таблиці 3.1 показані відомості про ресурси, використані в FPGA.

Таблиця 3.1 – Ресурси, які використані в FPGA

Число CLBс	3471/600000 (3%)
Число bonded IOBs	201/ (%)
Число глобальних буферів	1/8 (12%)
Сума використаних елементів	3948
Мінімальний період	25.405 ns
Максимальна частота	39.362 MHz
Максимальна затримка на лінії	1.148 ns
Середня затримка зв'язку	3.494 ns
Average Connection Delay on critical nets	0.000 ns
Середнє розфазування синхронізуючих імпульсів	0.248 ns
Максимальна затримка контакту	1.148 ns
Середня затримка зв'язку на гірших зв'язках	9.184 ns

Навіть центральний процесор високої швидкодії не може здійснити навчання НМ з великою кількістю нейронів і синапсів, а також формувати відповідь мережі в режимі реального часу. Реалізована апаратно НМ може проводити арифметичні операції набагато швидше, в порівнянні з її програмною реалізацією. FPGA обробляють дані, що поступають паралельно з високою швидкістю, що скорочує загальний час обчислень. Окрім цього, архітектура FPGA достатньо проста і може бути без істотних витрат розширена до 128 входів.

До недоліків реалізованої апаратно НМ слід віднести такі чинники як складна схема синхронізації при складних імплементаціях. Окрім цього, кількість вхідних/вихідних ліній обмежена ресурсами плати. Апаратна реалізація НМ не має гнучкості, властивої програмним засобам і для неї важко реалізувати процес навчання мережі.

У даному підрозділі продемонстрована методологія проектування заздалегідь навченої НМ для вирішення задачі прогнозування. Оцінені апаратні і тимчасові витрати.

НМ має цифрову реалізацію, що базується на мові VHDL, яка забезпечує ефективне використання площі кристала і швидкий автоматизований технологічний процес проектування [56].

Результати дипломної роботи плануються використовуватися на ТзОВ «Тетел», що підтверджує довідка про використання (додаток А).

Отже, в даному розділі показана ефективність використання розроблених методів на конкретних прикладах діагностики персонального комп'ютера.

Розглянутий процес апаратного синтезу заданої нейронної мережі у вигляді програмованої логічної матриці. Описана архітектура нейронної мережі, що синтезується. Приведений опис синтезованої моделі.

ВИСНОВКИ

1. Проаналізовані існуючі підходи до діагностики комп'ютерних систем за допомогою експертних систем. Цей аналіз показує, що найбільш поширеною формою представлення знань експерта є продукційні системи у поєднанні з іншими формами представлення знань (моделі, міркування на основі досвіду, фрейми і т.д.). Важливим недоліком продукційних діагностичних експертних систем є те, що існуючі методи їх побудови не гарантують діагностику довільного стану досліджуваного ними об'єкту. Для діагностики комп'ютерних систем в реальному часі це досконало неприпустимо.

2. Проаналізовані сучасні методи експертної класифікації і прийняття рішень, які гарантують постановку діагнозу будь-якому стану об'єкту дослідження. Цей аналіз показує, що при великій розмірності вектора, що описує стан таких складних об'єктів як комп'ютерна система або мережа, ці методи складні навіть для досвідченого експерта і результати можуть бути помилкові і суперечливі. Тому актуальним є використання продукційних систем (як найбільш прийнятної форми представлення знань експерта в процесі діагностики комп'ютерних систем і мереж) у поєднанні з методами експертної класифікації, що гарантують постановку діагнозу будь-якому стану об'єкту дослідження.

3. Проаналізовані сучасні методи і технології штучного інтелекту. Цей аналіз показує, що важливим напрямом розвитку систем штучного інтелекту є інтеграція нейронних мереж з продукційними правилами. Тому актуальною є апаратна реалізація нейронних мереж, яка забезпечить максимальну швидкодію при вирішенні задач діагностики комп'ютерних систем в реальному часі на основі продукційних експертних систем, що гарантують постановку діагнозу будь-якому стану об'єкту дослідження.

4. Розглянуто процес аналізу продукційних експертних систем, які здатні поставити діагноз будь-якому стану комп'ютерної системи, що характеризуються одним, двома, трьома і довільним числом параметрів. Показано, що використання бінарних відносин характерності значень параметрів для діагнозів дозволяє істотно скоротити число продукцій, необхідних для діагностики довільного стану

комп'ютерної системи.

5. Запропонований процес постановки діагнозу, який заснований на продукційному висновку і використанні бінарних відносин характерності значень параметрів для діагнозів. Використовуючи бінарні відносини характерності значень параметрів для діагнозів можна мінімізувати кількість продукцій із збереженням можливості постановки діагнозу будь-якому стану комп'ютерної системи.

6. Досліджена ефективність використання розроблених методів на конкретних прикладах діагностики персонального комп'ютера.

7. Розглянутий процес апаратного синтезу заданої нейронної мережі у вигляді програмованої логічної матриці. Описана архітектура нейронної мережі, що синтезується. Приведений опис синтезованої моделі.

СПИСОК ВИКОРИСТАНИХ ДЖЕРЕЛ

1. Гаврилова Т.А., Хорошевский В.Ф. Базы знаний интеллектуальных систем. – СПб: Питер, 2000. – 384 с.
2. Уотермен Д. Руководство по экспертным системам: Пер. с англ. – М.: Мир, 1989. – 388 с.
3. Addis T.R. Expert systems: an evolution in information retrieval. International Computers Limited (ICL) Technical Journal, May 1980.
4. Hartley R.T. CRIB: computer fault-finding through knowledge engineering. Computer, vol. 17, no. 3, March 1984.
5. Bennett J.S., Hollander C.R. DART: an expert system for computer fault diagnosis. Proceedings IJCAI-81, pp. 843-845, 1981.
6. Shubin H., Ulrich J.W. IDT: an intelligent diagnostic tool. Proceedings AAAI-82, pp. 290-295, 1982.
7. Kornell J. A VAX tuning expert built using automated knowledge acquisition. Proceedings of the First Conference on Artificial Intelligence Applications, IEEE Computer Society, December 1984.
8. Laffey T.J., Perkins W.A., Fischein O. LES: a model-based expert system for electronic maintenance. Proceeding of the Joint Services Workshop on AI in Maintenance, October 4-6, 1984, pp. 1-17.
9. Kelly V.E., Steinberg L.I. The CRITTER system: analyzing digital circuits by propagation behaviors and specifications. Proceedings AAAI-82, pp. 284-289, 1982.
10. Horstmann P.W. Design for testability using logic programming. Proceedings of the 1983 international Test Conference, IEEE Computer Society Press, Silver Spring, Md., 1983.
11. Sussman G.J., Holloway J., Knight T. Computer aided evolutionary design for

- digital integrated systems. Proceedings of the 1980 AISB Conference, 1980.
12. Alexander J.H., Freiling M.J. Building an expert system in SMALLTALK-80. Technical Report No. CR-85-06, Artificial Intelligence Dept., Computer Research Laboratory, Tektronix, Inc., November 1984.
 13. Finin T., McAdams J., Kleinosky P. FOREST: an expert system for automatic test equipment. Proceedings of the First Conference on Artificial Intelligence Applications, IEEE Computer Society, December 1984.
 14. Stefik M., Bobrow D., Bell A., Brown H., Conway L., Tong C. The partitioning of concerns in digital system design. Proceedings of the Conference on Advanced Research in VLSI, MIT, 1982.
 15. Люгер Д.Ф. Искусственный интеллект: стратегии и методы решения сложных проблем, 4-е издание: Пер. с англ. – М.: Издательский дом «Вильямс», 2003. – 864 с.
 16. Clancy W.J. Heuristic Classification. *Artificial Intelligence*, 27: 289-350, 1985.
 17. Davis R. and Hamscher W. Model-based Reasoning: Trouble Shooting. In Hamscher W., Console L. and deKleer J. *Readings in Model-based Diagnosis*. San Mateo: Morgan Kaufmann, 1992.
 18. Brown J.S., Burton R.R. and deKleer J. Pedagogical, natural language and knowledge engineering techniques in SOPHIE. In Sleeman D. and Brown J.S. *Intelligent Tutoring Systems*. New York: Academic Press, 1982.
 19. Stern C.R. and Luger G.F. Abduction and abstraction in diagnosis: a schema-based account, *Situated Cognition: Expertise in Context*. Ford et al. ed. Cambridge, MA: MIT Press, 1997.
 20. Skinner J.M. and Luger G.F. An architecture for integrating reasoning paradigms. *Principles of Knowledge Representation and Reasoning*. B. Nobel, C. Rich and W. Swartout, ed. San Mateo, CA: Morgan Kaufmann, 1992.

21. Chilukuri K.M. *Frontiers of expert system: reasoning with limited knowledge*. – Norwell: Kluwer Academic Publishes, 2000. – 297 p.
22. Keravnou E.T., Johnson L. *NEOGRIB: an Expert Fault Finding System that Articulates the Competence of Field Engineers* // 3 Int. Conf. FTCS. – Bremerhaven, Sept. 1987. – P.107-118.
23. Ларичев О.И. *Теория и методы принятия решений*. – М.: Логос, 2002. – 392 с.
24. Солсо Р. *Когнитивная психология*. – СПб: Питер, 2002. – 592 с.
25. Мошкович Е.М. *Конструктивный поиск и устранение противоречий в предпочтениях лица, принимающего решения при разбиении многомерных альтернатив на конечное число классов* // Проблемы и процедуры принятия решений при многих критериях. Сборник трудов / Под ред. С.В. Емельянова, О.И. Ларичева. – М.: ВНИИСИ, 1982. – № 6. – С. 73-80.
26. Ансель Ж. *О числе монотонных булевых функций от n переменных* // Кибернетический сборник. Н. с. – М.: Мир, 1968. – № 5. – С. 53-57.
27. Алексеев В.Б. *О расшифровке некоторых классов монотонных многозначных функций* // Журн. вычисл. математики и мат. физики. – 1976. – Т. 16, №1. – С. 189-198.
28. Соколов Н.А. *Об оптимальной расшифровке монотонных функций алгебры логики* // Журн. вычисл. математики и мат. физики. – 1982. – Т. 22, № 2. – С. 449-461.
29. Соколов Н.А. *Оптимальная расшифровка монотонных булевых функций* // Журн. вычисл. математики и мат. физики. – 1987. – Т. 27, № 12. – С. 1878-1887.
30. Ларичев О.И., Болотов А.А. *Система ДИФКЛАСС: Построение полных и непротиворечивых баз экспертных знаний в задачах дифференциальной классификации* // НТИ. Сер. 2, Информ. процессы и системы. – 1996. – № 9. – С. 9-15.

31. Асанов А.А. Методы извлечения и анализа экспертных знаний: Дис. ... канд. техн. наук. – М., 2002. – 129 с.
32. Кочин Д.Ю. Метод классификации заданного множества многокритериальных альтернатив // Методы поддержки принятия решений: Сб. тр. Ин-та систем. анализа РАН / Под ред. О.И. Ларичева. – М.: Эдиториал УРСС, 2001. – С. 4-18.
33. Ларичев О.И., Зуев Ю.А., Гнеденко Л.С. Метод ЗАПРОС (ЗАмкнутые Процедуры у Опорных Ситуаций) анализа вариантов сложных решений // Многокритериальный выбор в слабоструктуризованных проблемах / Под ред. С.В. Емельянова. – М.: ВНИИСИ, 1978. – С. 83-97.
34. Larichev O.I., Moshkovich H.M. ZAPROS-LM: A method and a system for ordering multiattribute alternatives // *Europ. J. Operat. Res.* – 1995. – Vol. 82, N3. – P. 503-521.
35. Larichev O.I. Ranking multicriteria alternatives: The method ZAPROS III // *Europ. J. Operat. Res.* – 2001. – Vol. 131, N 3. – P. 550-558.
36. Larichev O., Brown R. Numerical and verbal decision analysis used for the problems of resources allocation in Arctic // *J. Multi-Criteria Decision Anal.* – 2000. – Vol. 9, N 6. – P. 263-274.
37. Larichev O.I., Kochin D.Yu., Ustinovicus L.L. Multicriteria method for choosing the best object for investments // *DSS in the Uncertainty of the Internet age.* – Katowice: The Karol Adamiecki Univ. of Econ., 2003. – P. 255-270.
38. Бондарев В.Н., Аде Ф.Г. Искусственный интеллект. – Севастополь: Изд-во СевНТУ, 2002. – 615 с.
39. Джонс М.Т. Программирование искусственного интеллекта в приложениях. – М.: ДМК Пресс, 2004. – 312 с.
40. Kohonen T. *Self-organizing Maps.* – Berlin: Springer-Verlag, 1995. – 363 p.
41. Эволюционные вычисления и генетические алгоритмы. Обзор прикладной

- и промышленной математики. Выпуск 5.– М.: "ТВП".– Т.3.– 1996.– 204 с.
- 42.Обработка нечеткой информации в системах принятия решений / А.Н. Борисов, А.В. Алексеев, Г.В. Меркурьев. – М.: Радио и связь, 1989. – 304 с.
- 43.Поспелов Д.А. Ситуационное управление: теория и практика. – М.: Наука, 1986. – 288 с.
- 44.Джексон П. Введение в экспертные системы. – М.: "Вильямс", 2001. – 624 с.
- 45.Sycara P.K. Multiagent Systems // AI MAGAZINE. – 1998. – V. 19. – № 2. – P. 79-93.
- 46.Marco Dorigo, Vittorio Maniezzo, Alberto Coloni. The Ant System: Optimization by a colony of cooperating agents. // IEEE Transactions on Systems, Man and Cybernetics – Part B, Vol. 26, No.1, 1996. – P. 1-13.
- 47.Кривуля Г.Ф., Липчанский А.И., Механна Сами, Зидат Хабис. Диагностика компьютерных сетей с использованием экспертных систем // Вестник ХГТУ, 2004. – №1(19). – С. 11-16.
- 48.Хаханов В.И. Техническая диагностика элементов и узлов персональных компьютеров. – К.: ИСМО, 1997. – 308 с.
- 49.Нессер Д.Дж. Оптимизация и поиск неисправностей в сетях. – К.: "Диалектика", 1996. – 384 с.
- 50.М. Дэвид Стоун, Альфред Пур Ваш РС. Проблемы и решения: Практ. пособ. / Пер. с англ. – М.: Издательство ЭКОМ, 2002. – 416 с.
- 51.Гук М.Ю. Аппаратные средства IBM РС. Энциклопедия. 3-е изд. – СПб.: Питер, 2006. – 1072 с.
- 52.Кнышев Д.А., Кузелин М.О. ПЛИС фирмы "Xilinx": описание структуры основных семейств. – М.: Издательский дом «Додэка-XXI», 2001.
- 53.Сулима Т.Н., Лесовик У.И. Использование метода нейронных сетей для прогнозирования состояния здоровья подростков по результатам медико-

- генетического обследования// Проблемы бионики. – Харьков, №57, 2002. – С. 61-66.
- 54.“IEEE Standard VHDL Language Reference Manual”, ANSI/IEEE Std 1076-1993.
- 55.Sjoholm S. and Lindh L. (1997). “VHDL for Designers”, Prentice-Hall, 1997, ISBN 0-13-473414-9
- 56.Липчанский А.И., Лесовик У.И., Зидат Хабис Синтез заданной нейронной сети в программируемую логику // Радиоелектроніка. Інформатика. Управління, 2004. – №1(11). – С. 122-127.
- 57.Методичні рекомендації до виконання дипломної роботи з освітньо-кваліфікаційного рівня “Магістр”. Спеціальність „Комп’ютерні системи та мережі” / О.М. Березький, Р.Б. Трембач, Г.М. Мельник / Під ред. О.М. Березького – Тернопіль: ТНЕУ, 2012.– 42 с.

Додаток А
Довідка про використання