Міністерство освіти і науки України
Харківський національний університет радіоелектроніки

Факультет _____комп'ютерних наук_____
(повна назва)

Кафедра _____Програмної інженерії_____
(повна назва)

# АТЕСТАЦІЙНА РОБОТА
# Пояснювальна записка

_____другий (магістерський)_____
(рівень вищої освіти)

**Дослідження моделей категорного аналізу при побудові логічних мереж**
(тема)

Виконав: студент 2 курсу, групи ІПЗм-18-4__
спеціальності _121- Інженерія програмного забезпечення_____
(код і повна назва спеціальності)
освітньо-наукової програми _Інженерія програмного_забезпечення_____
(повна назва освітньої програми)

_____Ковалько М.С._____
(прізвище, ініціали)

Керівник _____проф. Дудар З.В._____
(посада, прізвище, ініціали)

Допускається до захисту

Зав. кафедри, проф. _____ З.В.Дудар

2020 р.

Харківський національний університет радіоелектроніки

Факультет комп'ютерних наук

Кафедра програмної інженерії

Рівень вищої освіти другий (магістерський)

Спеціальність 121– Інженерія програмного забезпечення
(код і повна назва)

Освітньо-наукова програма Інженерія програмного забезпечення
(повна назва)

ЗАТВЕРДЖУЮ:
Зав. кафедри _____
(підпис)
«____»_____ 20 ___ р.

# ЗАВДАННЯ
## НА АТЕСТАЦІЙНУ РОБОТУ

Студентові Ковалько Максиму Сергійовичу
(прізвище, ім'я, по батькові)

1. Тема роботи Дослідження моделей категорного аналізу при побудові логічних мереж

затверджена наказом по університету від « » 2020 р № ___

2. Термін подання студентом роботи до екзаменаційної комісії «10» травня 2020 р.

3. Вихідні дані до роботи *Алгоритми обробки алгебро-логічних рівнянь, алгоритми побудови логічних мереж, пояснювальна записка. Використовувати ОС Windows, середовище об'єктно-орієнтованого проектування.*

4. Перелік питань, що потрібно опрацювати в роботі *мета роботи, аналіз проблемної галузі і постановка задачі, методи пошуку корисних даних, опис об'єктних моделей, використовувані. методи та алгоритми, архітектура програмної системи, опис розробленої програмної системи. , результати тестування програмної системи*

5. Консультанти розділів роботи

| Найменування розділу | Консультант (посаду, прізвище, ім’я, по батькові) | Позначка консультанта про виконання розділу | |
|---|---|---|---|
| | | підпис | дата |
| Спецчастина | проф. Дудар З.В. | | |

## КАЛЕНДАРНИЙ ПЛАН

| № | Назва етапів роботи | Терміни виконання етапів роботи | Примітка |
|---|---|---|---|
| 1. | Аналіз предметної галузі | 25 березня 2020 р. | |
| 2. | Огляд існуючих методів | 31 березня 2019 р. | |
| 3. | Методи побудови логічних мереж | 15 квітня 2020 р. | |
| 4. | Підготовка пояснювальної записки | 20 квітня 2020 р. | |
| 5. | Спецчастина | 28 квітня 2020 р. | |
| 6. | Підготовка презентації та доповіді | 03 травня 2020 р. | |
| 7. | Попередній захист | 05 травня 2020 р. | |
| 8. | Нормоконтроль, рецензування | 07 травня 2020 р. | |
| 9. | Занесення диплома в електронний архів | 08 травня 2020 р. | |
| 10. | Допуск до захисту в зав. кафедри | 10 травня 2020 р. | |
| | | | |

Дата видачі завдання _ «____»_ _____2020 р.

Студент _____
(підпис)

Керівник роботи _____ __проф. Дудар З.В._____
(підпис) (посада, прізвище, ініціали)

# РЕФЕРАТ ABSTRACT

Пояснювальна записка до атестаційної роботи: 66 с, 8 табл., 35 рис., 3 дод., 25 джерел.

АЛГЕБРА ПРЕДИКАТІВ, ПРЕДИКАТНА КАТЕГОРІЯ, ЛОГІЧНА МЕРЕЖА, МОЗКОПОДІБНА ЕОМ.

Об'єктом розробки є категорні моделі логічних мереж мозкоподібних ЕОМ.

Методом розробки вибрано модифіковані та предикатні категорії, методи побудови логічних мереж.

Метою роботи є розробка алгоритмів логічних мереж та програмної моделі логічної мережі і застосування її до актуальних завдань моделювання штучного інтелекту у галузі обробки неформалізованої інформації.

У результаті роботи побудовано класифікацію завдань з цієї галузі. Розглянуто математичні методи представлення інформації, побудовано програмну модель логічної мережі.

ALGEBRA OF PREDICATES, NORMALIZATION, ANALYSIS, CATEGORY OF PREDICATES, LOGICAL NETWORK, BRAINLIKE COMPUTER

The methods of development are the modified and predicate categories, methods of logical networks construction.

The purpose of work is the development of universal logical network programmatic model and application of it to the actual tasks of artificial intelligence from industry of not formalized information treatment.

As a result of work classification of tasks from this industry is built, programmatic model of logical network is conducted.

.

# ЗМІСТ

Вступ..... ……6
1 Аналіз стану розв'язання проблеми та обґрунтування цілей дослідження …….10
1.1 Аналіз методів створення мозкоподібних комп’ютерів ……...10
1.2 Математичний апарат теорії інтелекту……...12
1.3 Аналіз методів предикатного опису логіки ……...15
1.4 Постановка завдання дослідження ……17
2 Опис об'єктної моделі, методів і алгоритмів ……19
2.1 Аналіз безоб'єктних категорій ……19
2.2 Алгоритми категорії з об'єктами ……22
2.3 Рішення предикатних рівнянь за допомогою логічних мереж ……25
3 Опис розроблених алгоритмів моделювання ……29
3.1 Алгоритм додавання проміжних змінних ……29
3.2 Модель алгоритму вибору варіантів мережі ……33
3.3 Розробка алгоритмів моделювання логічної мережі ……36
4 Опис розробленого програмного продукту .....38
4.1 Опис програмної реалізації ……38
4.2 Програмна модель процесу рішення завдання ……39
5 Опис можливості використання отриманих результатів.....43
Висновки ..... ……47
Перелік джерел посилання ..... ……48
Додаток А Програмний код..... ……49
Додаток Б Слайди презентації …….....53
Додаток В Апробація результатів роботи……65

# ВСТУП

Бурхливий розвиток комп'ютерних і інформаційних технологій привів у наші дні до різкого росту кількості збереженої й оброблюваної інформації. Як наслідок – з'являється необхідність автоматизації її обробки. Зокрема, усе більш важливими й актуальними стають завдання обробки – розпізнавання, синтезу й нормалізації такої інформації, що погано формалізується (найбільш природної й зручної для безпосереднього сприйняття людиною) – зображень, аудіо й відеосигналів, інформації природною мовою. Більшість із них, не дивлячись на свою актуальність, не мають на даний момент загального рішення [1]. Це обумовлене, зокрема, двома основними причинами:

– трудомісткістю моделювання спільних завдань внаслідок складності формального опису й недостатньої розробки необхідних для такого опису формалізмів;

– високими вимогами до апаратних ресурсів обчислювальних систем для рішення таких завдань;

Друга науково-технічна революція поставила перед наукою завдання автоматизації аспектів інтелектуальної діяльності людину, що привело до розвитку кібернетики, інформатики й, зокрема, теорії інтелекту. Так, наприклад, ріст обсягів текстової, зокрема – науково-технічної документації (сучасні бази даних містять десятки мільйонів документів) зробив актуальним завданням пошуку необхідної інформації й привів до появи множини робіт на цю тему [2]. Очевидно, зір також можна віднести до процесів, що ставляться до інтелектуальної діяльності [3] – «очі – частина мозку, винесена на периферію». Таке формулювання дозволяє говорити про можливість застосування досягнень теорії штучного інтелекту й у завданнях комп'ютерного зору.

На прикладі проблем комп'ютерного зору можна розглянути загальну класифікацію завдань виникаючих у процесі обробки неформалізованої інформації й розв'язуваних штучним інтелектом:

– завдання аналізу (сюди ставиться, наприклад, розпізнавання – одержання з неформалізованої інформації конкретних параметрів, необхідних для застосування деякого формалізму);

– завдання нормалізації – приведення інформації до деякої еталонної форми, що особливо актуально в завданнях пошуку інформації;

– завдання синтезу – вираження внутрішньої технічної вистави інформації, збереженої відповідно до формальних вимог у вигляді, адаптованому для сприйняття людиною;

– змішані завдання – найбільш загальний і трудомісткий клас завдань.

Серед множини формалізмів, застосовних тією чи іншою мірою до завдань обробки неформалізованої інформації, найбільш доцільним представляється використання алгебри предикатів, системи рівнянь якої реалізуються технічно у вигляді логічної мережі. Однією з переваг такого підходу є його пряма застосовність до всіх з перерахованих у класифікації типам завдань, включаючи останній, що забезпечується декларативністю логічної мережі як методу рішення систем предикатних рівнянь. Запис же самих предикатних рівнянь стає можливим завдяки алгебрі предикатів.

Іншою перевагою логічних мереж є широке розпаралелювання обчислень, що за умови правильної побудови моделі завдання дає гарантію високої ефективності. На сьогоднішній день розпаралелювання широко застосовується в окремих завданнях штучного інтелекту, зокрема – при розпізнаванні образів. Однак, побудова ЕОМ паралельної дії проводиться в рамках окремих конкретних завдань [4]. ЕОМ паралельної дії, що працює по принципах мозку й побудована на сучасній елементній базі, згідно із приблизними оцінками, буде перевершувати по продуктивності нинішні ЕОМ послідовної дії в 1012 раз, а мозок людини – в 106 разів [1].

Крім того, мозкоподібна ЕОМ, заснована на технології логічних мереж, представляється близької по будові до структури людського мозку, що дає підставу сподіватися на створення заснованих на цій технології систем обробки інформації, близьких по можливостях до людських, які по своїх характеристиках

помітно перевершують усі існуючі на даний момент системи, що мають відношення до штучного інтелекту.

З появою й розвитком штучного інтелекту його розроблювачі все частіше обертають свої погляди до теорії категорій, знаходячи в ній дуже важливі для себе ідеї. Основи теорії категорій були закладені на початку 400х років XX сторіччя. Вивчення категорій швидко перетворилося в самостійну абстрактну дисципліну, і тепер становить важливу галузі чистої математики. Крім того, вона вплинула на понятійні основи математики й мова математичної практики. Вона пропонує елегантні й потужні засоби для виразу зв'язків між великими галузями математики й постачає математиків знаряддями математичного дослідження, що займають усе більше й більше місця в арсеналі математики.

Існуючі теорії з'являються в новім світлі після переформулювання їх у теоретико-стрілочних термінах. Теорія категорій досяглася значних успіхів у виявленні й розробці основних, потужних математичних понять. І тепер вона пропонує новий теоретичний каркас для самої математики.

Скористатися на практиці досягненнями теорії категорій інженерові, що працює в галузі інформатизації, непросто. Теорія категорій і інформатизація – це далекі друг від друга області знання. У публікаціях фахівціввматематиків по теорії категорій ні слова не говориться про прив'язку її змісту до потреб інформатизації. Тому на сьогоднішній день теорія категорій ще мало допомагає вирішувати такі проблеми, як опис механізмів людського інтелекту, як розвиток штучного інтелекту, створення ЕОМ паралельної дії, що працюють за принципом мозку, називаних мозкоподібними ЕОМ і т.п.

У зв'язку з вищесказаним виникла проблема – як зробити теорію категорій застосовної до прикладних завдань із різних галузей інформатизації, як відмобілізувати багатство змісту теорії категорій на службу інформатизації. Спроби застосувати теорію категорій в інформатизації почалися з 90-х років XX сторіччя. Теорія категорій розглядалася як спосіб вивчення системних об'єктів [5]; засобу теорії категорій використовувалися в області математичного опису баз даних [6].

Метою атестаційної роботи є встановити зв'язок між теорією категорій і інформатизацією, розробити програмну систему, що моделює поведінку мозкоподібних комп’ютерів. У якості проміжної галузі знань використовується алгебра кінцевих предикатів, засобами якої була побудована предикатна інтерпретація категорії. На базі цієї інтерпретації розроблено елементи теорії модифікованої категорії для процесів моделювання функцій штучного інтелекту.

# 1 АНАЛІЗ СТАНУ РОЗВ'ЯЗАННЯ ПРОБЛЕМИ ТА ОБҐРУНТУВАННЯ ЦІЛЕЙ ДОСЛІДЖЕННЯ

## 1.1 Аналіз методів створення мозкоподібних комп’ютерів

Відмінною рисою розглянутих завдань штучного інтелекту є те, що вони надзвичайно легко й природно вирішуються розумом людини й дуже складно піддаються формалізації, необхідної для того, щоб уможливити автоматизацію їх рішення. Наприклад, в області пошуку інформації дотепер не вирішене завдання автоматичної побудови індексу документа для пошукових систем орієнтованих на точність результатів. У такому ж стані перебувають і необхідні моделі [2]–[3]. Таким чином, зовсім природно виглядає спроба досліджувати універсальний інструмент обробки неформалізованої інформації – мозок людину й змоделювати деякі його функції, що стосуються поставлених завдань. Природно, підхід до моделювання функцій мозку повинен опиратися на існуючі досягнення технічних наук і математики, для того, щоб результати моделювання могли бути технічно реалізовані на базі сучасних технологій. Суть підходу полягає в тому, що інтелект людини розглядається як логіка в дії, як деяке матеріальне втілення механізму логіки. Були виконані роботи з алгебраїзації логіки [4]. У результаті розроблений спеціальний математичний апарат для формульної вистави відносин і дій над ними, які називаються алгебро-логічними структурами [5]–[6]. Відносини інтерпретуються як думки інтелекту, а дії над ними – як мислення. Схемна реалізація формул, що описують алгебро-логічні структури, приводить до характерних інженерних мереж, що не використовувалися раніше, які називаються логічними мережами. Кожний тип алгебро-логічних структур (а таких типів порівняно небагато) приводить до свого типу логічних мереж з легко пізнаваної на око схемою. При зіставленні цих типів мереж з основними типами нейроструктур виявляється глибока подібність будови технічних і біологічних конструкцій. Опираючись на цю подібність, можна визначити функції різних типів нейронних структур і описати в точних математичних і технічних термінах

принципи функціонування мозку. Головне в даному методі – це рух зверху вниз: від загальних системних міркувань до алгебро-логічним структурам, а від них – до логічних мереж, які потім ототожнюються з біологічними нейронними структурами. Фахівці ж по нейрокомп'ютерах сьогодні намагаються йти іншим шляхом: від біологічних нейронних мереж до принципів їх дії, а від них – до інженерних рішень. Цей шлях приводить до значних труднощів через недолік знань про функції біологічних нейронних структур [3]. Рухаючись зверху вниз, ми приходимо до єдино прийнятних принципів побудови нейронних структур мозку людину. Згорнути убік тут немає ніякої можливості. У результаті вимальовуються прості і ясні принципи побудови мозкоподібних ЕОМ. Вони суттєво відрізняються від усього того, що дотепер використовувалося при паралельної обробки інформації, зокрема – при створенні ЕОМ паралельної дії. Основу мозкоподібних комп’ютерів становлять логічні мережі.

Дослідження, проведені в даній предметній області, привели до розробки багатьох логічних понять і підходів. Деякі з них є вже невід'ємними для будь-якої теорії, що має відношення до штучного інтелекту – це поняття відносини – як єдиного математичного об'єкта придатного для формального запису будь-якої інформації, предиката – функціонального еквівалента відносини й інші. Загальним результатом досліджень з'явилося усвідомлення необхідності декларативного підходу при моделюванні завдань штучного інтелекту. У відмінності від традиційного для інформаційних технологій алгоритмічного підходу, декларативна модель завдання не являє собою алгоритму. Модель завдання представлена в декларативній формі звичайно являє собою набір фактів, знань про факти й правил, згідно з якими проводяться дії над фактами, що дозволяють одержати нові знання на основі старих. Однак, навіть у сучасних розробках декларативний підхід застосовується тільки на самому верхньому рівні. Сучасні декларативні мови реалізують алгоритмічно представлену з їхньою допомогою модель. Таким чином, досягається широта описових можливостей, властива декларативному підходу, але зовсім не використовуються можливості паралельних обчислень, які цей підхід надає. Дійсно, множина дій виконуються

незалежно друг від друга, крім того, самі дії являють собою елементарні логічні операції, легко реалізовані на сучасній технологічній базі.

Логічна мережа являє собою метод рішення систем предикатних рівнянь, і має описові можливості алгебри предикатів [7]. З іншої сторони логічна мережа легко відображається у відповідне електронне обладнання, легко реалізоване на основі сучасних технологій. Властивістю логічної мережі є паралельне виконання всіх елементарних логічних операцій – тобто максимально можливе на логічному рівні паралельних обчислень. Такий підхід гарантує високу ефективність рішення завдань, що зводяться до логічного висновку або до рішення логічних рівнянь.

## 1.2 Математичний апарат теорії інтелекту

Відносини виражають властивості предметів і зв'язки між ними. Вони являють собою універсальний засіб опису будь-яких об'єктів. За дві з половиною тисячі років розвитку науки не вдалося виявити у світі жодного об'єкта, про який можна було б із упевненістю сказати, що він, у принципі, не піддається опису за допомогою відносин [6]. Ніякі інші відомі засоби опису об'єктів (наприклад, шкільна алгебра, інтегральне вирахування) властивістю універсальності не мають. Мова людини, що представляє собою головний засіб духовного спілкування людей, можна розглядати лише як механізм для вираження відносин. Звертаючись із мовою до інших людей, ми передаємо їм цілком певний зміст вимовного пропозиції, який є ніщо інше, як деяке відношення. Обмін думками між людьми здійснюється тільки за рахунок передачі й приймання відносин. Кожна думка являє собою якесь відношення. Наукове визначення поняття відносини сформував у першій половині XVII століття Р. Декарт, спираючись на введені їм поняття змінної й координатної системи. З їхньою допомогою вводиться поняття простору. Відносини – це всілякі частини деякого простору.

Нижче дається розгорнута характеристика поняття відносини. Довільно виберемо яку-небудь непусту множина *U*, елементи якого будемо називати предметами. Сама множина *U* називається універсумом предметів. Воно може бути як кінцевим, так і нескінченним. Приклади універсумів предметів: множина {а, б, …, я} усіх букв російського алфавіту; множина {0, 1} булевих елементів; множина $N$={1, 2,…} усіх натуральних чисел; множина *R* усіх дійсних чисел; множина усіх функцій, що відображають *R* в *R*. Довільно виберемо m яких-небудь непустих необов'язково різних підмножин $A_1, A_2,\ldots, A_m$, універсума *U*. Декартовий добуток $S = A_1, A_2,\ldots, A_m$ множин $A_1, A_2,\ldots, A_m$ називається предметним простором *S* з координатними осями $A_1, A_2,\ldots, A_m$ над універсумом *U*. Число осей m називається розмірністю простору *S*.

Уведено множина V={$x_1, x_2,\ldots, x_m$} різних змінних $x_1, x_2,\ldots, x_m$, які називаються предметними змінними простори *S*. Множина V називається універсумом змінних простору *S*. Значеннями змінної $x_i$ ($i=\overline{1,m}$) служать елементи множина $A_i$, так що $x_1 \in A_1$, $x_2 \in A_2,\ldots, x_m \in A_m$. Деякі з множинаей у переліку $A_1, A_2,\ldots, A_m$ можуть збігатися один з одним.

Множина $A_1, A_2,\ldots, A_m$ називаються областями завдання змінних $x_1, x_2,\ldots, x_m$. Кожною зміною *xi* ($i=\overline{1,m}$) ставиться у відповідність якась фіксована область завдання *Ai*. У результаті множина $A_1, A_2,\ldots, A_m$ утворюють набір ($A_1, A_2,\ldots, A_m$). Ми обмежилися введенням кінцевого универсума змінних *V*. Однак, його можна брати й нескінченним. У першому випадку простір називається скінченим, а в другому – безкінечним. Просторів може бути багато. Кожне з них має свої універсум предметів, універсум змінних і набір областей завдання змінних. Іноді виникає необхідність використовувати систему з декількох взаємозалежних просторів, які можуть відрізнятися друг від друга, мати різну розмірність. У цьому випадку додатково треба ввести загальні для всіх таких просторів універсум предметів і універсум змінних, які повинні охоплювати универсуми предметів і змінних усіх просторів, що входять у цю систему.

Змінні можна змістовно інтерпретувати як місця, а їх значення (тобто предмети) як стану цих місць. Приклади місць і їх станів: температура людини в

різних точках його тіла; колір зорового відчуття в різних точках поля зору: швидкість автомобіля в різні моменти часу; спектральна щільність променистої яскравості світлового випромінювання в різних точках його спектра ( тобто при різних довжинах хвиль електромагнітних коливань). Той факт, що місце x перебуває в стані a (можна сказати й інакше: на місці x розташований предмет a), записується у вигляді рівності $x = a$. Як зовнішній (об'єктивний), так і внутрішній (суб'єктивний) мири людини влаштовані одноманітно: усюди ми виявляємо лише місця, які в кожний момент часу перебувають у певних станах [8]. Із часом стани місць можуть мінятися. Не може так трапитися, щоб у деякому місці в якийсь момент часу не було ніякого стану, щоб на заданім місці не містився який-небудь предмет. Неможливо також, щоб кілька предметів займали те саме місце. Якщо якийсь предмет займає деяке місце, то ніякий інший предмет там перебувати вже не може. Буває, що в одній крапці простору виявляється одночасно багато станів. У цьому випадку, щоб уникнути логічних протиріч, їх усе доводиться поєднувати в один векторний стан, виділяючи для кожного його компонента своє місце. Так, наприклад, окремі спектральні лінії у світловім випромінюванні поєднують в один стан, називаний його спектром. Термін «місце» розуміється в узагальненому змісті. Не обов'язково його розуміти як просторову крапку. Місцем, приміром, може бути деякий момент часу, оцінка на шкалі частот, одне з полів шахової дошки.

Якщо $x_1 \in A_1$, $x_2 \in A_2, \ldots$, $x_m \in A_m$ і $x_1 = a_1$, $x_2 = a_2, \ldots$, $x_m = a_m$, то пишуть $a_m$ і $(a_1, a_2, \ldots, a_m) \in S$ і говорять, що предметний вектор $(a_1, a_2, \ldots, a_m)$ (або інакше: набір предметів $a_1, a_2, \ldots, a_m$) належить простору $S = A_1, A_2, \ldots, A_m$. Елементи $a_1, a_2, \ldots, a_m$ вектора $(a_1, a_2, \ldots, a_m)$ називаються його компонентами (першим, другим,…, $m$-тим). Простір $S$ можна розглядати як сукупність усіх векторів виду $(x_1, x_2, \ldots, x_m)$, кожний з яких задовольняє умові $x_1 \in A_1$, $x_2 \in A_2, \ldots$, $x_m \in A_m$. Будь-яка підмножина $P$ простору $S$ називається відношенням, утвореним в (або інакше: заданим на) просторі $S$. Відношення має розмірність $m$. Говорять, що воно *m-місцево*. Відносини, задані на тому самому просторі $S$, називаються однотипними. Тип відносини визначається множиною $V$ і набором $(A_1, A_2, \ldots, A_m)$.

Розглянуто приклад відносини. Нехай $U$={1, 2, 3, 4, 5, 6}, $V$={$x_1$, $x_2$}, $m$=2, $A_1$={1, 2, 3, 4}, $A_2$={3, 4, 5, 6}. Простір $S$=$A_1$×$A_2$ є множина усіх пар виду ($x_1$, $x_2$), що задовольняють умові $x_1$∈$A_1$, $x_2$∈ $A_2$: $S$={(1, 3), (1, 4), (1, 5), (1, 6), (2, 3), (2, 4), (2, 5), (2, 6), (3, 3), (3, 4), (3, 5), (3, 6), (4, 3), (4, 4), (4, 5), (4, 6)}. Простір $S$ складається з 4×4= 16 двокомпонентних векторів. У ролі відносини $P$ беремо одне з підмножин простору $S$: $P$={(1, 6), (2, 4), (3, 3), (4, 3), (4, 4)}. Усього в просторі $S$ можна утворювати 216=65536 відносин. Відношення ∅, що не містить жодного вектора, називається порожнім, відношення $S$, у якім є всілякі вектори, – повним.

## 1.3 Аналіз методів предикатного опису логіки

У теорії інтелекту інтелект людини розглядається як логіка в дії, як деяке матеріальне втілення механізму логіки. У рамках цього наукового напрямку були виконані велика кількість робіт.

В [14] розроблено універсальний математичний апарат для формального опису дискретних об'єктів і явищ, який був названий алгеброю кінцевих предикатів. До таких об'єктів і явищ ставляться алгоритми, програми, схеми ЕОМ, граф-схеми, лінгвістичні й логічні закономірності, багато явищ соціології, економіки, біології, психології й генетики, механізми розпізнавання образів, інтелектуальні процеси. Описувані математичні засоби є узагальненням апарата алгебри логіки й багатозначної логіки. Використання алгебри предикатів дозволило почати формальний опис абстрактних понять, якими користується людина у своїй інтелектуальній діяльності.

Мова алгебри предикатів являє собою універсальний засіб формального опису будь-яких механізмів інтелекту людину й машини [15]. Алгебра предикатів – це перший щабель формальної мови для розроблювачів, що проектують засобу штучного інтелекту. Мовою алгебри предикатів виражаються будь-які відносини.

Відносини виражають властивості предметів і зв'язки між ними. Вони являють собою універсальний засіб формального опису будь-яких об'єктів. В [14] стверджується, що ніякі інші відомі засоби формального опису об'єктів (наприклад шкільна алгебра, диференціальне й інтегральне вирахування) властивістю універсальності не мають. Наприклад мовою формул алгебри предикатів, у принципі, можна записати будь-яке інтегральне рівняння. Мова інтегрального вирахування – це спеціалізована мова, мова ж алгебри предикатів – універсальний.

Предикати — це основний математичний інструмент, призначений для формального опису об'єктів біоніки інтелекту. Важливо мати у своєму розпорядженні математичні засоби запису відносин за допомогою формул.

Будь-яке відношення можна змістовно проінтерпретувати як знання про факт, виражене деяким висловленням. Факт – це вичерпна характеристика дійсного стану всіх місць, що цікавлять нас. Знання ж про факт не завжди має таку визначеність, воно лише обмежує множина можливих станів місць. Нічим іншим факти й знання бути не можуть. Проінтерпретуємо, приміром, цим способом записане вище відношення $P$. Є два нас, що цікавлять, місця $x_1$ та $x_2$. Відомо, що місце $x_1$ може перебувати тільки в станах 1, 2, 3, 4, а місце $x_2$ – тільки в станах 3, 4, 5, 6. Припустимо, що фактичні стани місць $x_1$ і $x_2$ спостерігачеві невідомі. Сам же факт об'єктивно характеризується парою (4, 3). Це означає, що місце $x_1$ перебуває в стані 4, а місце $x_2$ – у стані 1. Спостерігачеві доводиться задовольняти знанням про факт у вигляді відносини $P$={(1, 6), (2, 4), (3, 3), (4, 3), (4, 4)}, яке обмежує множина пар можливих станів місць із 16 до 5. Спираючись на це знання, спостерігач містить, що пари станів (1, 3), (1, 4), (1, 5), (2, 3), (2, 5), (2, 6), (3, 4), (3, 5), (3, 6), (4, 5), (4, 6) не мають місця в дійсності. Цю інформацію він може використовувати у своїх міркуваннях при виробленні яких-небудь рішень.

Аналіз одномісні й двомісних відносинах, з якими нам багаторазово прийде мати справа надалі. Якщо в записі одномісного відношення забрати круглі дужки,

то ми одержимо запис відповідного йому множина, наприклад, {(1), (3), (4)} = {1, 3, 4}.

Множина і відносини взаємно однозначно зв'язані один з одним, тому їх можна не розрізняти [7]. Надалі одномісні відносини для стислості можна записувати як множину. Двомісні відносини зручно наочно представляти у вигляді двочасткових графів. Нижче для прикладу зображений двочастковий граф відносини $P$={(1, 6), (2, 4), (3, 3), (4, 3), (4, 4)}. Кожній парі, що входить до складу відносини, відповідає своє ребро графа, яке з'єднує його вершини, позначені іменами компонентів цієї пари (рисунок 1.1).

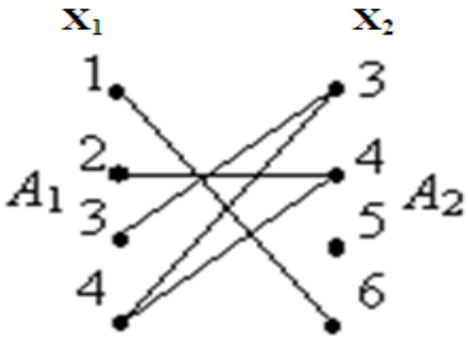


Рисунок 1.1 – Граф відносини

Це ж відношення можна представити у вигляді графіків і таблиць різного виду. У рамках підходу, досліджуваного в даній роботі, саме відносини (і взаємно однозначно відповідні їм предикати) використовуються для формальної вистави інформації

## 1.4 Постановка завдання дослідження

Для формалізації завдань обробки неформалізованої інформації за допомогою логічних мереж необхідно досліджувати можливості раціонального застосування алгебри кінцевих предикатів [8]–[11]. Із цією метою необхідно провести моделювання найбільш загальних завдань із різних галузей даної

предметної області. Як приклад обраний такі галузі як: завдання моделювання механізмів природної мови – зокрема, завдання відмінювання повних імен прикметників, завдання логічного розпізнавання й завдання комп'ютерного зору. Такий вибір зроблений у силу актуальності наведених галузей. Крім того, необхідно теоретично досліджувати властивості даного методу, виявити його гідності й недоліки й побудувати програмну модель логічної мережі, щоб одержати можливість проведення машинного експерименту й показати можливість реалізації підходу на ЕОМ. Слід також розглянути рішення практичних завдань на побудованих моделях логічних мереж за допомогою програмної моделі.

Таким чином, постановка завдання дослідження атестаційної роботи магістра зводиться до наступного:

– проаналізувати переваги і недоліки існуючих методів;

– формалізувати за допомогою алгебри предикатів і побудувати логічну мережу завдання логічного розпізнавання;

– досліджувати формалізацію за допомогою алгебри предикатів і моделювання логічною мережею рішення булевих рівнянь;

– узагальнити завдання логічного розпізнавання й побудувати загальну модель;

– побудувати універсальну програмну модель логічної мережі;

– побудувати програмні моделі розглянутих завдань і досліджувати їхню роботу за допомогою програмної моделі;

– універсальна програмна модель повинна задовольняти наступним вимогам:

– дозволяти введення й редагування моделей логічних мереж;

– моделювати роботу логічної мережі, при необхідності – по тактах;

– виводити на кожному такті інформацію про стан мережі;

– допускати можливість зміни стану мережі вручну між будь-якими двома тактами.

# 2 ОПИС ОБ'ЄКТНОЇ МОДЕЛІ, МЕТОДІВ І АЛГОРИТМІВ

## 2.1 Аналіз безоб'єктних категорій

Спочатку проведено аналіз класичних категорій, після чого здійснено її предикатну інтерпретацію. Аналіз одержуваної таким способом предикатної категорії виявляє деяку її анормальність, що дає привід для коректування поняття класичної категорії. Здійснивши таке коректування, ми приходимо до поняття модифікованої категорії, яке, як нам представляється, підходить на роль відправного пункту побудови теоретичної бази створення мозкоподібних ЕОМ. Далі здійснюємо розробку теорії модифікованих категорій. Щоб таку теорію можна було безпосереднє практично використовувати в справі створення мозкоподібних ЕОМ, необхідно дати її предикатну інтерпретацію.

Відмінність між теорією категорій і алгеброю предикатів полягає лише в тому, що перша здійснює рух зверху вниз, націлена на пізнання вищих логічних механізмів і тому використовує в якості відправних поняття рекордного рівня спільності. Друга ж, відправляючись від потреб інформатизації, рухається у вивченні тієї ж логіки мислення знизу нагору. Якщо б удалося дати переконливу інтерпретацію понять, формованих теорією категорій, і методів, розроблювальних нею, у термінах алгебри предикатів, тобто, конкретизуючи, наблизити їх до інформатизації, те це, по-перше, суттєво збагатило б інструментарій алгебри предикатів, по-друге, дало можливість донести багатство ідей теорії категорій до фахівців, що рухають уперед інформатизацію. А якби вдалося абстрагувати (узагальнити) конструкції алгебри предикатів, те це виявила б стимулююча дія з боку інформатизації на розвиток самої теорії категорій. Саме алгебру предикатів ми збираємося використовувати в ролі такої проміжної області знання.

Скористатися на практиці досягненнями теорії категорій інженерові, що працює в галузі інформатизації, непросто. Занадто великий розрив у рівні абстракції. Крім того, теорія категорій і інформатизація – це дуже далекі друг від друга області знання. У публікаціях фахівців математиків по теорії категорій ні

слова не говориться про прив'язку її змісту до потреб інформатизації. Можна було б суттєво полегшити рішення цього завдання, якщо б вдалося знайти проміжну область знання середнього рівня абстрактності, що зв'язує теорію категорій із практикою інформатизації, яка змогла б виконати роль посередника між ними.

Логічна мережа копіює дії людини, але з тією лише різницею, що людей при цьому діє послідовно, а мережа – паралельно. Мережа працює по тактах. Кожний такт ділиться на два півтакти – перший і другий. У першому півтакті *i*-го такту мережа для кожного зі своїх рівнянь виду $ДО(x, y)=1$ ( *ДО* –відношення, задане таким рівнянням) відшукує: 1) по відомим знанню $P_i(x)$ про значення змінної *x* на початку *i*-го такту знання $Q'_i(y)$ про значення змінної *в* наприкінці *i*-го такту; 2) по відомім знанню $Qi(y)$ про значення змінної *b* на початку *i*-го такту знання $P'_i(x)$ про значення змінної *x* наприкінці *i*-го такту. Математично ці дві операції виражаються формулами:

$$\exists x \in A(K(x, y)Pi(x))=Q'_i(y); \tag{2.1}$$

$$\exists x \in B(K(x, y)Qi(y))=P'_i(x). \tag{2.2}$$

де *A* и *B*– області зміни змінних *x* и *в*.

У другому півтакті кожного такту мережа відшукує загальну частину $Pi_{+1}(x)$ усіх знань $P'_{i1}(x)$, $P'_{i2}(x),\ldots, P'_{il}(x)$ про значення кожної зі своїх предметних змінних *x* сторін, що надходять *по* галузях мережі із усіх, до полюса x. Виражається ця операція в такий спосіб:

$$P'_{i1}(x) \wedge P'_{i2}(x) \wedge \ldots \wedge P'_{il}(x)=Pi_{+1}(x). \tag{2.3}$$

Отримане знання $Pi_{+1}(x)$ потім використовується в ролі стану полюса *x* у початковий момент *i*+ *1г*го такту. Символ *l* позначає число галузей, що підходять до полюса *x*. До початку *i*+ 1гго такту в кожному полюсі формується знання о множині $Pi_{+1}(x)$, яка завжди виявляється включеним у знання о множині $Pi(x)$, що втримувалася в тому ж полюсі на початку *i*-го такту. Таким чином, єдиним результатом роботи логічної мережі є уточнення знань, що втримуються у всіх її полюсах відповідно до вихідних даних.

Мережа вирішується з використанням потоків, кожний потік прив’язується до певного полюса й перетинає всі пов'язані з ним галузями полюса на підставі

таблиці відносин відповідної до галузей. У процесі перетинання створюється безліч перетинання на підставі таблиці відносин галузі, і стані пов'язаним з потоком полюса, після чого проводиться логічне множення між безліччю перетинання й безліччю стану пересічного полюса, і результат записується в стан полюса. У процесі рішення відбувається синхронізація потоків, запускаються тільки ті потоки, стан полюсів яких змінився після останнього запуску потоків. Якщо стан жодного з полюса потоків уже не змінюється, усі потоки зупиняються.

Основною областю застосування даного програмного продукту є розробка логічних мереж і їх тестування, для наступного застосування в системах пов'язаних зі штучним інтелектом, і створення швидкодіючих карт, також окремі модулі програми можуть використовуватися розроблювачами ПС для створення систем штучного інтелекту на основі бінарних логічних мереж.

Система може бути дороблена в напрямку виключень обмежень, а також можуть бути додані модулі для експорту мережі у відомі СУБД.

При впровадженні теорії категорій поряд зі звичайним поняттям категорії зустрілося й більш загальне поняття безоб'єктної категорії.

Нехай M – деяка множина. Її елементи, позначувані символами *f, g, h,…,* називаються морфізмами. Нехай, крім того, задана якась, в загальному випадку, часткова, операція $fg = h$, що діє з $M \times M$ у $M$. Вона називається множенням морфізмів $f$ і $g$. Морфізм $h$ називається добутком морфізмів $f$ і $g$. Будь-який морфизм $e \in M$ називається одиничним або тотожним, якщо він задовольняє наступним умовам:

– $ee=e$;

– при будь-якому $f \in M$, для якого існує добуток $fe \in M$, виконується рівність $fe=f$;

– при будь-якому $f \in M$, для якого існує добуток $ef \in M$, виконується рівність $ef=f$.

Цим визначенням мовчазно допускається існування в категорії багатьох одиниць. Саме наявність багатьох одиниць (і тільки це) відрізняє, по великому рахункові, категорію від інших відомих алгебраїчних структур. Одиничний

елемент був би один, якщо б на множині M операція множення була не часткової, а всюди певної. Одиничні морфізми $ef$ і $e'_f$ називаються відповідно до правих і лівим для морфізма $f \in M$, якщо $fef=f$ і $fe'_f=f$. Для будь-якого морфізма $f \in M$ існують єдиний правий і єдиний лівий одиничні морфізми. Це твердження називається законом тотожності. Таким чином, для кожного $f \in M$ існує єдина права одиниця $ef$ і єдина ліва одиниця $e'_f$, такі, що $fef = fe'_f = f$.

Множення морфизмов асоціативно: $(fg)\,h = f(gh)$ при будь-яких $f, g, h \in M$, для яких існують добутку: $(fg)h$ і $f(gh)$. Застереження про існування добутків $(fg)h$ і $f(gh)$ була б зайвою при всюди певній операції множення морфизмов. Добуток $fg$ морфизмов $f$ і $g$ існує в тому й тільки тому випадку, коли правий одиничний морфизм морфизма $f$ збігається з лівим одиничним морфизмом морфизма $g$. Таким чином, необхідною й достатньою умовою існування добутку $fg$ морфизмов $f$ і $g$ є наявність такого одиничного морфизма $e$, що $fe=f$ і $eg=g$. Множина $M$, у якім утримуються, принаймні, одиничні морфизми, і на якім задана операція множення морфизмов із зазначеними вище властивостями називається безоб'єктною класичною категорією $K$. Пишуть $M$=Mork, $f \in M$, $f \in$Mork. Mork – це множина усіх морфизмов категорії $K$. Якщо $f \in$Mork, то говорять, що морфизм $f$ є $K$-Морфізмом.

## 2.2 Алгоритми категорії з об'єктами

У наведенім вище визначенні категорії морфізми були представлені поки дуже схематично – лише як безструктурні елементи деякої множинаі. Конкретизуємо поняття категорії й морфизма. У процесі конкретизації раніше введене поняття обростає додатковими властивостями. На додаток до морфизмам категорії $K$ уводимо об'єкти категорії $K$. Множина усіх об'єктів категорії $K$ записується у вигляді Ob в $K$ або Obk. Об'єкти позначаємо буквами $A, B, C,\ldots$ Якщо $A \in$Obk, то говорять, що $A$ є $K$-об'єктом. Говорять, що $f$ є морфизм із об'єкта

$A$ в об'єкт $B$, і пишуть $f{:}A{\to}B$ або $A\xrightarrow{f} B$. Об'єкт $A$ називається початком морфизма $f$, а об'єкт $B$ – його кінцем. Замість терміна «морфизм» також використовується слово стрільця.

Кожній парі ($A$, $B$) об'єктів $A$, $B{\in}$Obk ставиться у відповідність якесь, можливо й порожнє, множина $HK(A, B)$ морфизмів категорії $K$. Можливий випадок, коли багатьом різним морфизмам, наприклад, $f$, $g$, $h$, поставлена у відповідність та сама пара об'єктів ($A$, $B$), тобто $f$, $g$, $h$: $A{\to}B$. Такі морфизми називаються паралельними. А для якоїсь іншої пари об'єктів ($C$, $D$) у категорії $K$ взагалі може не найтися жодного морфізма $f$, такого що $f$: $C{\to}D$. Замість запису $HK(A, B)$ також використовуються позначення Hom($A$, $B$), Mork($A$, $B$), $K(A, B)$, а якщо це не приводить до двозначності, – те й більш лаконічні записи $H(A, B)$, Hom($A$, $B$), Mor($A$, $B$). Замість запису $f{\in}HK(A, B)$ інакше пишуть $f$: $A{\to}B$ або $A\xrightarrow{f} B$. Замість виражень «об'єкт $A{\in}$Obk» і «морфизм $f{\in}$Mork» пишуть «об'єкт $A{\in}K$» і «морфизм $f{\in}K$» або ще простіше: « К*к**Об'єкт A*» і « К-*Морфизм f*». Для кожного морфизма $f{\in}$Mork існує єдина пара об'єктів $A$ і $B$, така що $A$, $B{\in}$Obk і $f{\in}HK(A, B)$. Приписування цієї властивості морфизмам мотивується тим, що для кожної функції $f$ треба обов'язково вказати її область визначення $A$ і область значень $B$, інакше визначення функції буде незавершеним. Пишуть $A$ = domf (початок морфизма, у прийнятій нами інтерпретації – його область визначення), $B{=}codf$ (кінець морфизма, у нашій інтерпретації – його область значень).

Категорія з об'єктами $K$ складається з множини морфизмов Mork і множини об'єктів Obk. Передбачається, що множини Mork і Obk не перетинаються. Категорія $K$ характеризується наступними п'ятьма властивостями:

– кожній парі $K$-об'єктів $A$, $B$ відповідає множина $HK(A, B)$ морфизмов ( можливо, навіть порожнє), включене в Mork;

– для кожного морфізма $f{\in}$Mork існує єдина пара $A$, $B$ *K-об'єктів*, така що $f{\in}HK(A, B)$;

– у множині Mork визначена, загалом кажучи часткова, двомісна операція – множення морфизмов; добуток $fg$ морфизмов $f{:}A{\to}B$ і $g{:}C{\to}D$ визначене лише в

тих випадках, коли $B=C$, тобто коли кінець морфизма $f$ збігається з початком морфизма $g$. У цьому випадку добуток $fg$ є K-*морфизм* із об'єкта $A$ в об'єкт $D$. У цьому випадку говорять, що для об'єктів $A, B, C \in K$ визначене відображення

$$H_K(A,B)\times H_K(B,C)\to H_K(A,C).$$

Знак × у цьому випадку позначає декартово добуток множинаей морфизмов. Морфизми $f$, $g$ категорії $K$ виду $f$: $A\to B$ і $g$: $B\to C$ називаються послідовними, а виду $f$: $A\to B$ і $g$: $A\to B$ – паралельними.

Множення морфизмов асоціативно:

$$(fg)h = f(gh) \qquad (2.4)$$

щораз, коли морфизми (fg)$h$ і $f$(gh) існують. Іншими словами, ассоциативность слушна щораз, коли $f$: $A\to B$, $g$: $B\to C$, $h$: $C\to D$. Таким чином, асоциативність виконується у всіх тих випадках, коли вона має сенс. Рівність (2.1) виражає категорний закон асоціативності.

Закон асоціативності можна наочно виразити графічно у вигляді категорной діаграми зображеної на рисунку 2.1.

Будь-яка категорна діаграма утворюється з об'єктів і стрілок (морфізмів), вона являє собою орієнтований граф з розфарбованими вершинами й дугами. У ролі вершин графа в категорної діаграмі виступають об'єкти категорії, а в ролі дуг

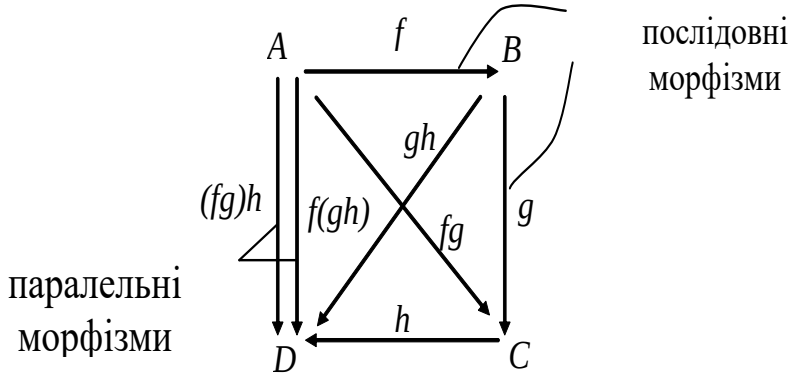


Рисунок 2.1 – Графічне подання закону асоціативності

її морфизми. Такого виду діаграми широко використовуються в теорії категорій. Вони – головний засіб наочної вистави внутрішньої будови й властивостей математичних структур, зв'язків між ними.

## 2.3 Рішення предикатних рівнянь за допомогою логічних мереж

Розглянуто на прикладі операції диз'юнкції принципи побудови й роботи логічної мережі при рішенні булевих рівнянь (більш простий приклад, операція заперечення – це вже бінарна мережа). Нехай задана операція

$$x \vee y = z .$$

Таблиця 2.1 Значення змінних логічної мережі

| $x$ | $y$ | $z$ | $t$ |
|---|---|---|---|
| 0 | 0 | 0 | 0 |
| 0 | 1 | 1 | 1 |
| 1 | 0 | 1 | 2 |
| 1 | 1 | 1 | 3 |

Логічна мережа має вигляд (рисунок 2.2):

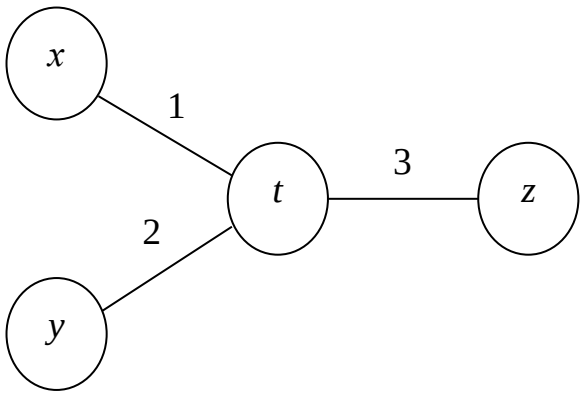


Рисунок 2.2 – Логічна мережа для обчислення операції диз'юнкції

Двочасткові графи зв'язків змінних мають вигляд (рисунок 2.3):

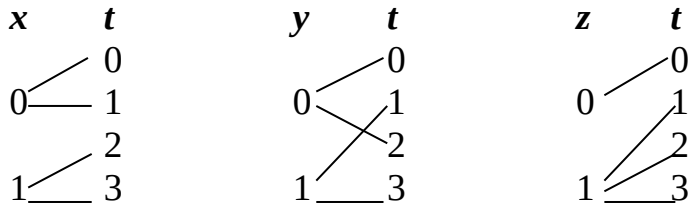


Рисунок 2.3 – Двочасткові графи зв'язків логічної мережі

Розглянуто приклади роботи мережі. Функція $x \vee y = z$ обчислюється однозначно по набору значень (*x, y)*. Однак у зворотну сторону – від значень змінної *z* до значень *x, в* однозначності немає. Щоб за значенням *z* знайти *x* і *y*, треба уточнити значення додаткової змінної *t*.

Аналітично зв'язок усіх змінних мережі запишеться так: $x^0 y^0 z^0 = t^0$.

На наборі *x* = 0; *в* = 0 розглянуто оба півтакти роботи мережі.

На першому півтакті задані значення кожної змінної *x*, *y* породжують множини відповідних їм значень змінної *t* – відповідно {0, 1} і {0, 2}. На другому півтакті обчислюється перетинання цих множин.

Одержано однозначне значення змінної *z*=0.

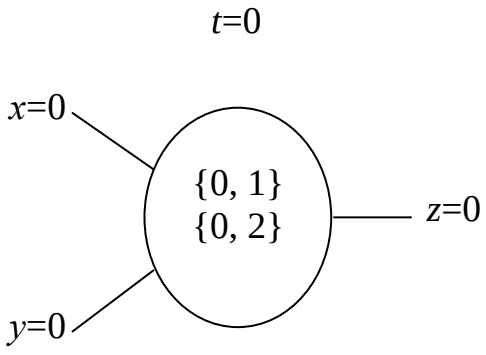


Рисунок 2.4 – Результат роботи логічної мережі на наборі *x*=0; *в*=0

При *x*=0; *в*=1, $x^0 y^1 z^1 = t^1$ мережа має такий вигляд (рисунок 2.5).

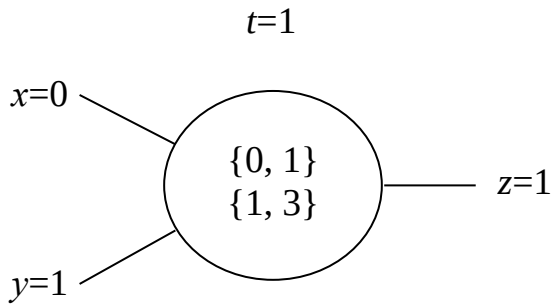


Рисунок 2.5 – Результат роботи логічної мережі на наборі $x$=0; $в$=1

При $x$=1; $в$=0, $x^1y^0z^1 = t^2$ Результат роботи мережі представлено на рис. 2.6.

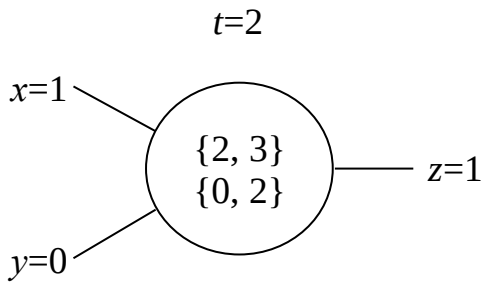


Рисунок 2.6 – Результат роботи логічної мережі на наборі $x$=1; $в$=0

При $x$ = 1; $в$ =1, $x^1y^1z^1 = t^3$ рис. 2.7.

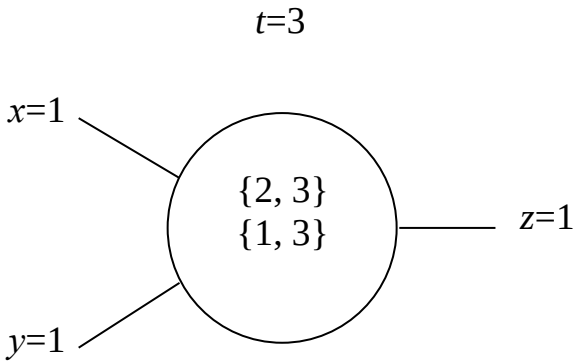


Рисунок 2.7 – Результат роботи логічної мережі на наборі $x$=1; $в$=1

Таким чином, виконано модель роботи мережі для всіх наборів значень *x*, *в*.

Будь-яка категорна діаграма утворюється з об'єктів і стрілок (морфізмів), вона являє собою орієнтований граф з розфарбованими вершинами й дугами.

У ролі вершин графа в категорній діаграмі виступають об'єкти категорії

Властивості математичного об'єкта (простору, групи, і т.п.), які зазвичай формулюються через його внутрішню структуру, досить ефективно виражаються через властивості відображень цього об'єкта в однотипні з ним об'єкти. Саме ця можливість – переводити вивчення внутрішньої структури у вивчення зовнішніх зв'язків пояснює роль теорії категорій у вивченні системних об'єктів.

# 3 ОПИС РОЗРОБЛЕНИХ АЛГОРИТМІВ МОДЕЛЮВАННЯ

## 3.1 Алгоритм додавання проміжних змінних

Вихідне відношення $S(x, y, z)$, що містить усю інформацію, що й дозволяє проводити зворотні обчислення, має вигляд:

$$S(x, y, z) = x^0 y^0 z^0 \vee x^0 y^1 z^1 \vee x^1 y^0 z^1 \vee x^1 y^1 z^1. \quad (3.1)$$

Використовуючи проміжну змінну $t$, одержимо повну (розгорнуту) бінарізацію:

$$R(x, y, z, t) = x^0 y^0 z^0 t^0 \vee x^0 y^1 z^1 t^1 \vee x^1 y^0 z^1 t^2 \vee x^1 y^1 z^1 t^3. \quad (3.2)$$

Результат безпосередньої бінарізації $S'(x, y, z)$ не збігається з вихідною диз'юнкцією $S(x, y, z)$. Цим доведене, що бінарізація неповна без проміжної змінної.

В роботі запропонований алгоритм, що представляє роботу схеми по тактах:

- затримка на один такт;
- перевірка на збіг попереднього й поточного сигналів;
- передача сигналів по інтерфейсу;
- зворотний зв'язок – передача сигналів по вертикальних ланцюгах;
- формування сигналів такту;
- перевірка на збіг поточного й наступного сигналів;
- зчитування результату.

За аналогією з побудованою логічною мережею для операції диз'юнкції можна побудувати мережі для будь-яких логічних операцій і використовувати їх у якості елементарних логічних блоків для моделювання й паралельної обробки будь-яких логічних відносин.

Розглянуто практично важливе завдання скорочення внутрішніх станів у логічній мережі (областей значень проміжних станів). Подібне завдання

розв'язали Хафмен і Милі [16], розробивши метод скорочення числа внутрішніх станів автомата.

Становиться таблиця функції диз'юнкції (буквену, а не логічну). В ній налічується три гнізда з одиницями. Це аналог однакових закінчень у парадигматичній таблиці відмінювання імен прикметників.

Таблиця 3.1 – Функція диз'юнкції

| $x$ / $y$ | 0 | 1 | |
|---|---|---|---|
| 0 | 0 | 1 | $z$ |
| 1 | 1 | 1 | |

Додавання проміжної змінної *t*:

$$x^0 y^0 = t^0;$$

$$x^0 y^1 = t^1;$$

$$x^1 y^0 = t^2;$$

$$x^1 y^1 = t^3.$$

Розглянуто всі варіанти виносу за дужки:

$$(x^0 \vee x^1) y^1 = t^1 \vee t^3;$$

$$x^1 (y^0 \vee y^1) = t^2 \vee t^3.$$

Інших варіантів склеювання конституент одиниці відносини диз'юнкції $S(x, y, z)$ немає.

Можна скоротити область зміни для змінної $t$ на один стан. Виходить два можливі варіанти областей:

$$M_1 = \{0,1,2\} \ (t_1^1 = t^1 \vee t^3, t_1^2 = t^2, t_1^0 = t^0 );$$

$$M_2 = \{0,1,3\} \ (t_2^0 = t^0, t_1^1 = t^1, \ t_1^2 = t^2 \vee t^3 ).$$

Одержано дві таблиці

Таблиця 3.2 – Таблиці для змінних $t_1$ і $t_2$

| $x$ | $y$ | $z$ | $t_1$ | $x$ | $y$ | $z$ | $t_2$ |
|---|---|---|---|---|---|---|---|
| 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 |
| 0 | 1 | 1 | 1 | 0 | 1 | 1 | 1 |
| 1 | 0 | 1 | 2 | 1 | 0 | 1 | 2 |
| 1 | 1 | 1 | 1 | 1 | 1 | 1 | 2 |

Первісна нумерація наборів наведена в табл. 3.3

Таблиця 3.3 – Таблиця для змінної $t$

| $x$ \ $y$ | 0 | 1 |
|---|---|---|
| 0 | 0 | 1 |
| 1 | 2 | 3 |

Перший варіант скорочення нумерації наведено в табл. 3.4::

Таблиця 3.4 – Таблиця зі склеєними гніздами для змінної $t_1$

| $x$ \ $y$ | 0 | 1 |
|---|---|---|
| 0 | 0 | 1 |
| 1 | 2 | 1 |

Другий варіант скорочення нумерації наведено у табл. 3.5:

Таблиця 3.5 – Склеєні гнізда для змінної $t_2$

| $x$ \ $y$ | 0 | 1 |
|---|---|---|
| 0 | 0 | 1 |
| 1 | 2 | 2 |

Немає необхідності вводити області зміни значень для предметних змінних, тому що коли використовуються формули для запису без заперечень, то з них уже можна витягати області зміни змінних. Треба відрізняти області зміни змінної, узятої поза зв'язком із предикатом, коли вона береться сама по собі, і область зміни змінної в складі предиката. Тоді предикат сам нав'язує область зміни для змінної.

Ця область визначається за допомогою квантора існування по іншим змінним предиката. Наприклад, $\exists y P(x, y) = M(x)$. $M$ – область зміни змінної $x$ щодо предиката $P$.

Перший варіант мережі:

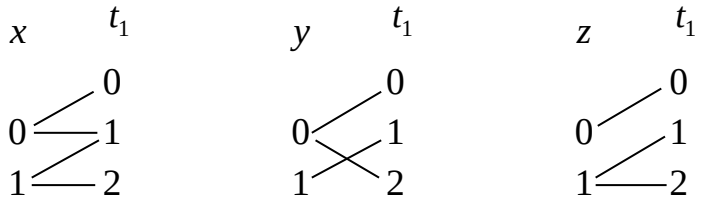


Рисунок 3.1 – Двочасткові графи зв'язків для змінної $t_1$

$$x^0 y^0 z^0 = t_1^0;$$

$$(x^0 \vee x^1) y^1 z^1 = t_1^1;$$

$$(y^0 \vee y^1) x^1 z^1 = t_1^2.$$

Другий варіант мережі (рисунок 3.2):

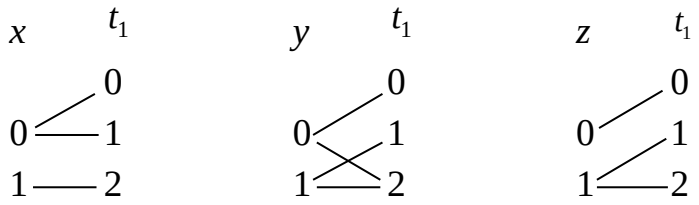


Рисунок 3.2 – Двочасткові графи зв'язків для змінної $t_2$

Алгоритм дозволяє побудувати мета-мережу для формули будь-якої булевої функції, також можна одержувати спосіб рішення булевих рівнянь, тобто одержано спосіб виводу логічних наслідків з посилок за допомогою мережі, що може бути реалізований програмно.

## 3.2 Модель алгоритму вибору варіантів мережі

Два варіанти, що розроблені є ізоморфними. Можна записати відношення ізоморфізму, що переводить один варіант в інший. Для побудови ізоморфізму потрібно:

- поміняти місцями змінні $x$ й $y$.
- для $t_1$ й $t_2$ поміняти місцями 1 та 2.

Ізоморфізм повністю характеризується табл. 3.6.

Таблиця 3.6 – Ізоморфізм змінних $t_1$ і $t_2$

| $t_1$ | $t_2$ |
|---|---|
| 0 | 0 |
| 1 | 2 |
| 2 | 1 |

Для представлення моделі завдання, що сформульовано в термінах алгебри кінцевих предикатів, у програмній моделі використовується формальна декларативна мова. Опис ділиться на дві частини – опис логічної мережі завдання – база даних, що містить правила, по яких можна одержати нові знання про факти, і база даних, що містить обмеження змінних (знання про факти)

Процес опису логічної мережі також ділиться на дві частини: опис змінних і опис відносин.

Опис змінних починається рядком `[Variables]`. При опису змінної необхідно задати її ім'я, тип і множина припустимих значень. Ім'я являє собою послідовність латинських букв і цифр не довше 20 символів. У програмній моделі передбачено 3 типу даних: ціле число, символ (буква або спеціальний символ за винятком роздільників) і рядок. Вимоги до даних останнього типу аналогічні вимогам іменування змінних. При завданні змінної вказується її ім'я, потім тип (I – ціле число c - символ *e* - рядок). Потім після символу "=" вказується множина значень змінної через пробіл. Наприкінці перерахування елементів множини ставиться знак ";". Таким чином, якщо змінна $x_1$ визначена на множині {М Ж С} це записується як «$x_1$ i = М Ж С ;».

Наведено як приклад опис змінних завдання відмінювання повних імен прикметників для української мови. Побудова моделі проводиться на основі парадигматичних таблиць відмінювання.

Вводяться необхідні предметні змінні: $x_1$ – рід форми слова зі значеннями М – чоловічий, Ж – жіночий, З – середній; $x_2$ – число форми слова зі значеннями *Е* – єдине, М – множинне; $x_3$ – падіж форми слова зі значеннями И – називний, Р – родовий, Д – давальний, В – знахідний, Т – орудний, П – прийменниковий; $x_4$ – ознака застосування форми слова зі значеннями З – сучасна, А – архаїчна; *t* – тип відмінювання слова зі значеннями 1-7; *z* – закінчення форми слова зі всіма можливими значеннями

```
[Variables]
x1 c = М Ж С;
x2 c = Е М;
x3 c = И Р Д У Т П;
x4 c = ПРО Н;
x5 c = З А;
```

```
r i = 1 2 3 4 5 6 7 8 9 10 11 12 13 14 15 16 17 18 19;
y1 c = БВДЗЛМПТЕНРСЖЧШЩЦГКХ;
y2 c = У Б;
y3 c = Т М;
s i = 1 2 3 4 5 6 7;
zл e = А А' Я В У' Ю ПРО О' Е И И' И И';
zп e = Й ГО МУ М Я Ю Е Х МІ;

z e = АЯ А'Я ЯЯ УЮ У'Ю ЮЮ ОЕ ОЙ ОМ ОГО ОМУ ОЮ О'Е О'Й О'М О'ГО О'МУ О'Ю ЇЇ
ЇЙ ЇМ ЙОГО ЙОМУ НЕЮ ИЙ ИМ ИЕ ИХ ИМИ И'М И'Е И'Х И'МИ ИЙ ЇМ ИЕ ЇХ НИМИ И'М
И'Е И'Х И'МИ;
z0 c = * 1;
z1 c = ИИЕУОЮЯА;
z2 c = ЙГМЯЮХЕ;
z3 c = ПРО*УИ;
```

Розділ опису відносин починається рядком [Relations]. Для опису відносини необхідно в круглих дужках указати імена змінних, які воно зв'язує. Після закриваючої фігурної дужки ставиться двокрапка. Потім у фігурних дужках вказуються пари множин, кожна з яких може бути образом і прообразом іншої. Множини в парі розділяються знаком рівності, після останнього елемента правої множини пари співставляється знак “;”. Наприклад, відношення, що задане предикатом

$$P_{15}(y_1, s) = (y_1^{Б} \vee y_1^{В} \vee y_1^{Д} \vee y_1^{З} \vee y_1^{Л} \vee y_1^{М} \vee y_1^{П} \vee y_1^{Т})(s^1 \vee$$
$$\vee s^6) \vee y_1^{Е} s^5 \vee (y_1^{Н} \vee y_1^{Р} \vee y_1^{С})(s^1 \vee s^5 \vee s^6) \vee (y_1^{Ж} \vee y_1^{Ч} \vee$$
$$\vee y_1^{Ш} \vee y_1^{Щ})(s^4 \vee s^7) \vee y_1^{Ц} s^3 \vee (y_1^{Г} \vee y_1^{К} \vee y_1^{Х})(s^2 \vee s^7).$$

Вираз в формі псевдокоду повинен бути записаним у вигляді

```
(s, y1):
{  1 6  = Б У Д З Л М П Т;
  5=Е;
  1 5 6=Н Р С;
  4 7=Ж Ш;
  4 = Ч Щ;
  3= Ц;
  7=Г ДО Х; }
```

Обмеження змінних будуються за тими ж правилами, що й опис змінних з тим лише відмінністю, що при вказівці обмежень не треба вказувати тип даних змінної.

Якщо необхідно обмежити множину значень змінної $x_1$ с {М Ж С} до {Ж З} це необхідно вказати в наступній формі: $x_1$ = Ж З; Значенням змінної $x_1$ стане перетинання зазначеної після знака рівності множини з областю її визначення.

## 3.3 Розробка алгоритмів моделювання логічної мережі

Процес роботи моделі логічної мережі ділиться на такти, кожний з яких складається із двох півтактів. Перед початком роботи мережа встановлюється у вихідний стан – уточнюються (зменшується потужність) множини завдання відповідних змінних, що відповідає появі нової інформації. Ця інформація вводиться користувачем у вікні «обмеження змінних» Змінні, яких торкнулися ці зміни, позначаються як модифіковані. На першому півтакті обчислюються образи множин, що відповідають зміненим змінним щодо ребер графа мережі, кожне з яких являє собою бінарний предикат.

Область істинності бінарного предиката завдає багатозначну функцію, значенням якої є підмножина області завдання змінній, відповідній до сусідньої вершини.

Наприклад, модифікована змінна $z_{\text{п}}$ має множину значень {М ГО}, а предикат, що зв'язує її з іншою змінною *r* має вигляд

$$P_7(z_{\text{п}}, r) = z_{\text{п}}^{\text{Й}}(r^1 \vee r^4 \vee r^9 \vee r^{11}) \vee z_{\text{п}}^{\text{ГО}}(r^2 \vee r^5) \vee z_{\text{п}}^{\text{МУ}} r^3 \vee$$
$$\vee z_{\text{п}}^{\text{М}}(r^6 \vee r^7 \vee r^{16}) \vee z_{\text{п}}^{\text{Я}} r^8.$$

Тоді її образом буде множина {2 5 6 7 16}.

Під час другого півтакту обчислюються перетинання знайдених на першому півтакті образів з множинами значень цільових змінних. Таким чином, якщо змінна *r* має множину значень {1 3 4 5 6 7 8}, то після другого півтакту ця множина скоротиться до {2 5 6 7 16} ∩ {1 3 4 5 6 7 8} = {5 6 7}. Ці перетинання являють собою нові значення цільових змінних.

Рішення завдання логічного розпізнавання у термінах мови програмної моделі завдання формалізується в такий спосіб:

```
[Variables]
A1 i = 1 0;  A2 i = 1 0; A3 i = 1 0;
K1 i = 1 0; K2 i = 1 0; K3 i = 1 0;
r i = 1 2 3 4 5 6 7 8;
[Relations]
  (r, A1):                          (K1, r):
  {    1 3 5 7 = 0;                 {    0 = 1 2 3;
```

```
     2 4 6 8 = 1; }
(r, A2):
{    1 2 5 6 = 0;
     3 4 7 8 = 1; }
(r, A3):
{    1 2 3 4 = 0;
     5 6 7 8 = 1; }
```

```
     1 = 6 8 5 7; }
(K2, r):
{    0 = 2 6 8;
     1 = 1 5 3 7; }
(K3, r):
{   0 = 1 5 6 7 8;
    1 = 2 3 7 8; }
```

Наступний етап алгоритму дозволяє визначити, який набір ознак гарантує, що в системі буде виконуватися процес ДО3, не буде виконуватися процес ДО2, якщо відомо, що ознака А1 має місце.

Таким чином, необхідно встановити наступні обмеження змінних:

```
-K3 = 1;
-K2 = 0;
-A2 = 1;
```

Якщо нові значення відмінні від старих, змінні позначаються як модифіковані й приймають участь у наступному такті.

Перед кожним тактом користувач має можливість одержати значення всіх змінних, або додати в мережу нову інформацію, скоротивши множину, відповідні до значень змінних.

Робота мережі триває доти, поки наприкінці кожного такту існують позначені змінні. Як тільки значення змінних перестають уточнюватися (мережа переходить у стабільний стан), мережа зупиняється і можливі три різні ситуації.

Якщо кожна змінна прийняла конкретне значення – стан мережі відповідає рішенню заданої системи предикатних рівнянь. Якщо деякі змінні зазначають множину значень – це говорить про те, що початкової інформації недостатньо для знаходження однозначного рішення і поставлене завдання допускає кілька рішень, що задовольняють поставленим умовам.

Випадок, коли значення всіх змінних відповідають порожнім множинам, свідчить про те, що початкові дані суперечливі.

# 4 ОПИС РОЗРОБЛЕНОЇ ПРОГРАМНОЇ СИСТЕМИ

## 4.1 Опис програмної реалізації

У ході атестаційної роботи була створена програмна модель логічної мережі, яка, не володіючи такими перевагами мозкоподібних комп’ютерів, як широкий паралелізм, все-таки надає можливості створення й налагодження моделей, проведення машинних експериментів і оцінки ефективності.

Для опису моделі мережі, використовується декларативна формальна мова, яка дозволяє задавати змінні (вузли) мережі й відносини - ребра. Головною необхідною властивістю програми є можливість одержувати й задавати стан мережі після кожного такту, що дозволяє стежити за ходом обчислень. У проекті наведений приклад опису логічної мережі для відмінювання повних неприсвійних імен прикметників мовою, використовуваному в створеній програмній моделі й результати роботи моделі мережі. Приклади показують ефективність роботи логічної мережі в умовах поставленого завдання й застосовність програмної моделі до рішення завдань штучного інтелекту.

Зовнішній вигляд інтерфейсу програми представлений на рис. 4.1 і надає наступні можливості:

– побудова опису моделі логічної мережі;

– завдання початкових умов для роботи мережі;

– збереження й завантаження цих описів;

– експлуатація логічної мережі – рішення систем рівнянь алгебри предикатів;

– по-тактове трасування роботи моделі для одержання наочної вистави про хід обчислень у мережі, пошуку й усунення помилок у моделі.

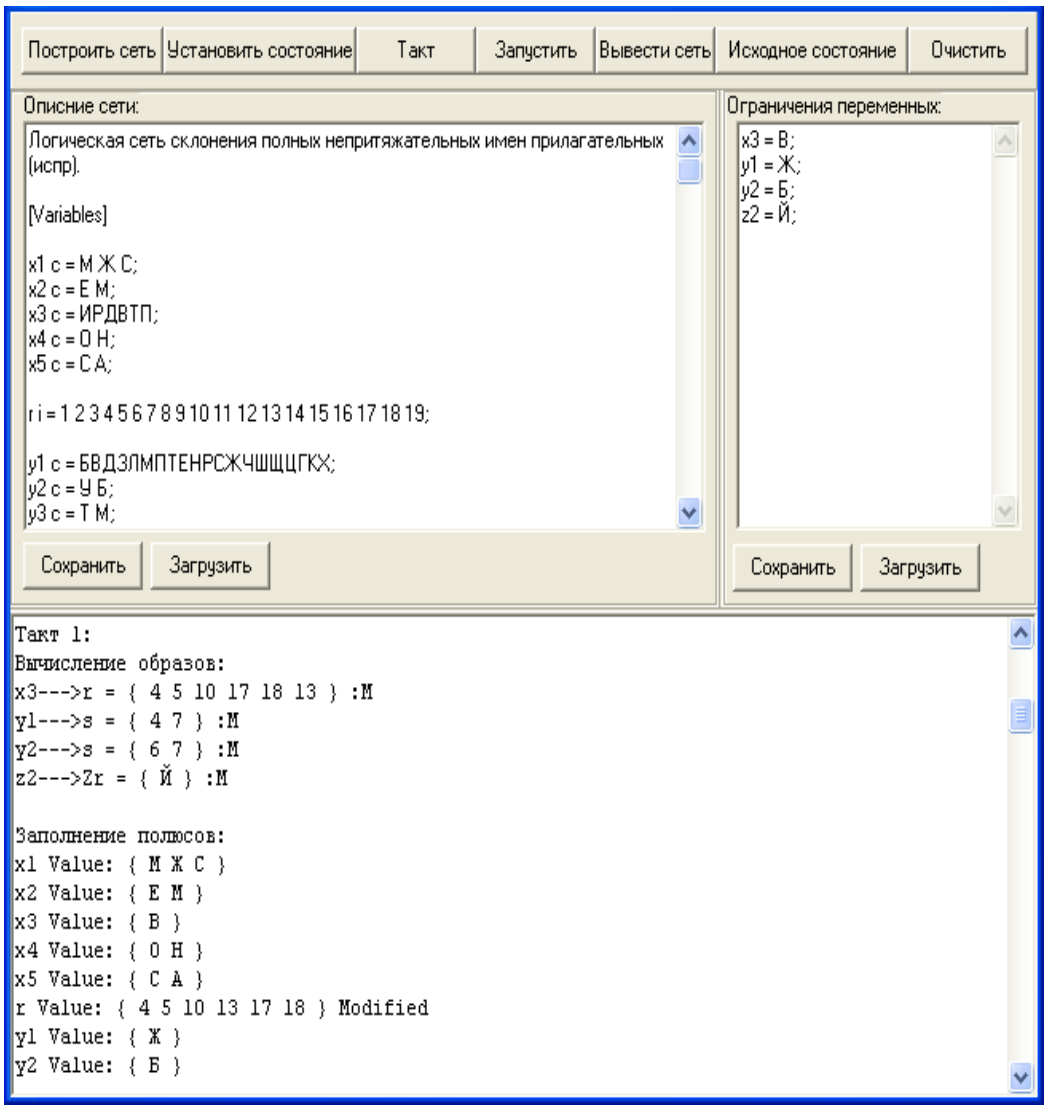


Рисунок 4.1 – Програмна модель логічної мережі

## 4.2 Програмна модель процесу рішення завдання

Результати виводяться в нижньому вікні форми. Спочатку вказується стан мережі перед початком роботи моделі. Для кожної змінної вказується множина її початкових значень. Змінені змінні позначені ключовим словом `Modified`.

Розглянуто результати роботи мережі на прикладі завдання відмінювання повних імен прикметників (рисунок 4.2). Нехай змінним були привласнені наступні значення:

$x_3$ = В И; $y_1$ = Ж;

$y_2$ = Б; $z_2$ = І;

```
Состояние сети перед началом работы:
x1 Value: { М Ж С }
x2 Value: { Е М }
x3 Value: { И В } Modified
x4 Value: { О Н }
x5 Value: { С А }
r Value: { 1 2 3 4 5 6 7 8 9 10 11 12 13 14 15 16 17 18 19 }
y1 Value: { Ж } Modified
y2 Value: { Б } Modified
y3 Value: { Т М }
s Value: { 1 2 3 4 5 6 7 }
Z1 Value: { А А` Я У У` Ю О О` Е Ы Ы` И И` }
Zr Value: { Й ГО МУ М Я Ю Е Х МИ }
Z Value: { АЯ А`Я ЯЯ УЮ У`Ю ЮЮ ОЕ ОЙ ОМ ОГО ОМУ ОЮ О`Е О`Й
О`М О`ГО О`МУ О`Ю ЕЕ ЕЙ ЕМ ЕГО ЕМУ ЕЮ ЫЙ ЫМ ЫЕ ЫХ ЫМИ Ы`М
Ы`Е Ы`Х Ы`МИ ИЙ ИМ ИЕ ИХ ИМИ И`М И`Е И`Х И`МИ }
z0 Value: { * 1 }
z1 Value: { Ы И Е У О Ю Я А }
z2 Value: { Й } Modified
z3 Value: { О * У И }
```

Рисунок 4.2 – Стан мережі перед початком роботи

Потім починається перший такт. Кожний такт складається із двох етапів – обчислення образів і заповнення полюсів (рисунок 4.3).

```
Такт 1:
Вычисление образов:
x3--->r = { 1 8 14 13 4 5 10 17 18 } :M
y1--->s = { 4 7 } :M
y2--->s = { 6 7 } :M
z2--->Zr = { Й } :M

Заполнение полюсов:
x1 Value: { М Ж С }
x2 Value: { Е М }
x3 Value: { И В }
x4 Value: { О Н }
x5 Value: { С А }
r Value: { 1 4 5 8 10 13 14 17 18 } Modified
y1 Value: { Ж }
y2 Value: { Б }
y3 Value: { Т М }
s Value: { 7 } Modified
Z1 Value: { А А` Я У У` Ю О О` Е Ы Ы` И И` }
Zr Value: { Й } Modified
Z Value: { АЯ А`Я ЯЯ УЮ У`Ю ЮЮ ОЕ ОЙ ОМ ОГО ОМУ ОЮ О`Е О`Й О`М О`ГО О`МУ О`Ю ЕЕ
ЕЙ ЕМ ЕГО ЕМУ ЕЮ ЫЙ ЫМ ЫЕ ЫХ ЫМИ Ы`М Ы`Е Ы`Х Ы`МИ ИЙ ИМ ИЕ ИХ ИМИ И`М И`Е И`Х
И`МИ }
z0 Value: { * 1 }
z1 Value: { Ы И Е У О Ю Я А }
z2 Value: { Й }
z3 Value: { О * У И }
```

Рисунок 4.3 – Перший такт

Образи обчислюються за правилами, зазначеними у підрозділі 3.2. Вирази “$x_3$ ---> $r$ = { 1 8 14 13 4 5 10 17 18 }: $M$” означає, що образом змінної $x_3$ щодо предиката, що зв'язує $x_3$ і $r$ буде множина { 1 8 14 13 4 5 10 17 18 }.

Позначка $M$ означає, що значення змінної $r$ буде змінено. Коли значення змінних перестають змінюватися виводиться повідомлення про то, що мережа стабільна й модель завершила роботу. На рис. 4.4 показаний стан мережі після останнього 5-го такту.

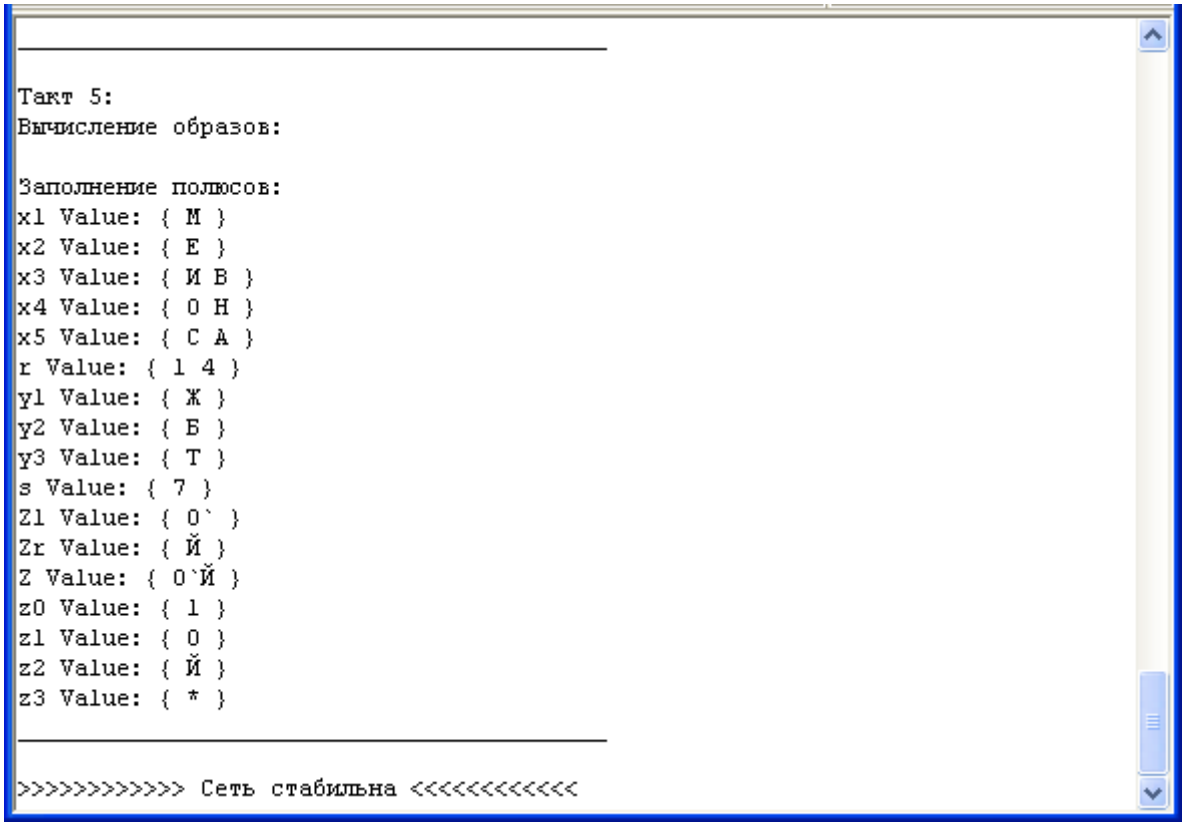


Рисунок 4.4 – Завершення роботи мережі

У термінах розв'язуваного завдання воно говорить про те, що досліджуваний прикметник чоловічого роду, знаходиться в однині. Із приводу натхненності нічого сказати не можна. Основа слова закінчується ненаголошеної твердої згодної. Слово має закінчення «ой», звук « про» - ударний.

Формалізація моделі логічної мережі для завдання, що представлено в п. 3.3. Далі розглянуто рішення в програмній моделі (рисунок 4.5).

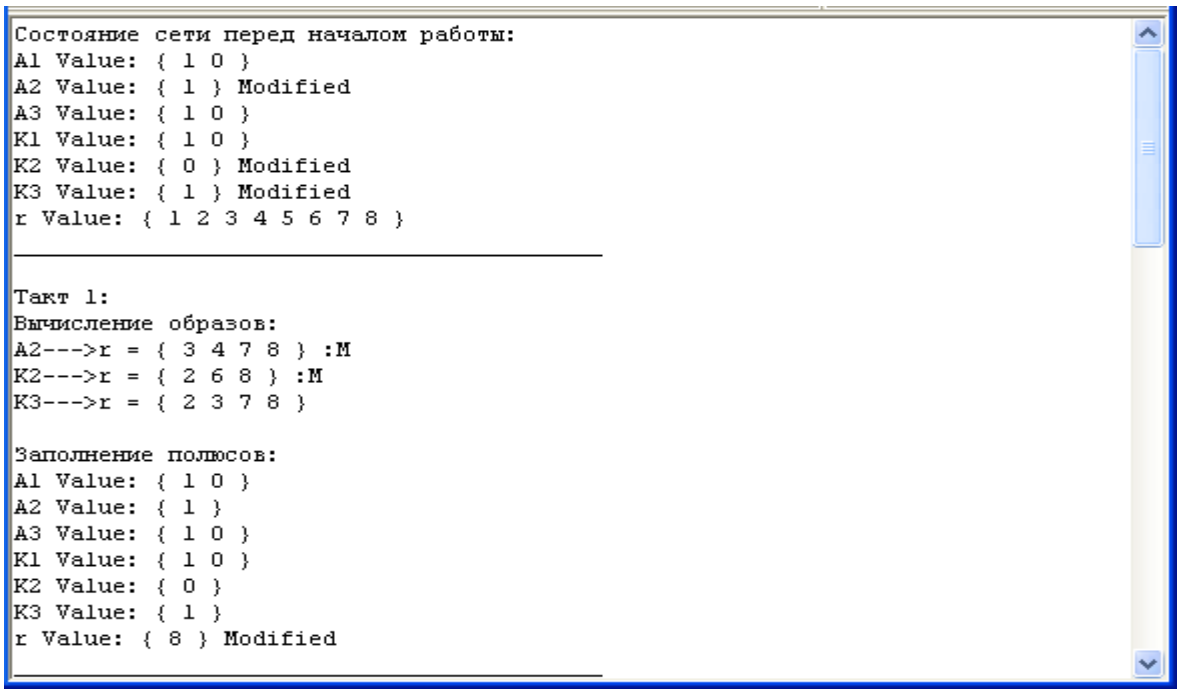


Рисунок 4.5 – Початок роботи моделі завдання логічного розпізнавання

На рис. 4.5 показано стан перед початком роботи моделі й перший такт. Обмежені (змінені) змінні впливають тільки на змінну *r*, оскільки не з'єднані мережею безпосередньо (рисунок 4.6).

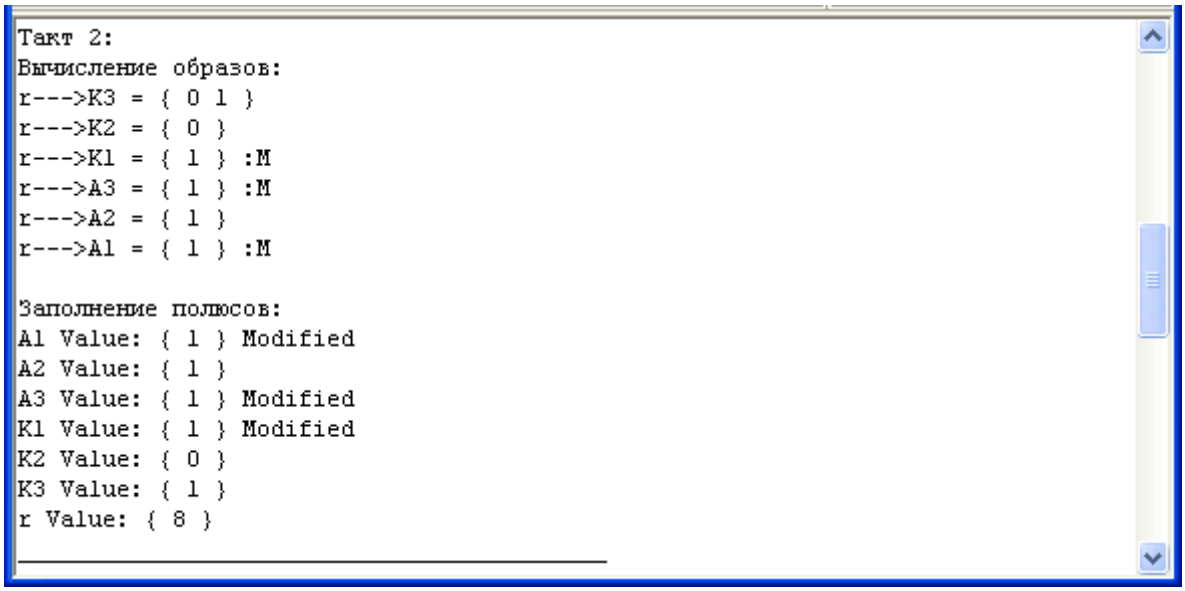


Рисунок 4.6 – Другий такт роботи моделі завдання логічного розпізнавання

Програмне рішення третього завдання, що розглянуто в проекті, – рішення булевих рівнянь, аналогічно першим двом.

# 5 ОПИС МОЖЛИВОСТІ ВИКОРИСТАННЯ ОТРИМАНИХ РЕЗУЛЬТАТІВ

Представлена програмна модель реалізує також можливість пошуку й усунення синтаксичних і логічних помилок у формальному описі моделі завдання. Так, наприклад, якщо в області визначення деякою зміною не присутня значення, яке вона може прийняти під час роботи моделі (воно зазначене у відношенні, яке зв'язує змінну з іншими), користувач одержить повідомлення про помилку. Можливість відслідковувати й коректувати стан мережі під час роботи моделі відкриває широкі можливості для проведення машинних експериментів (рисунок 5.1).

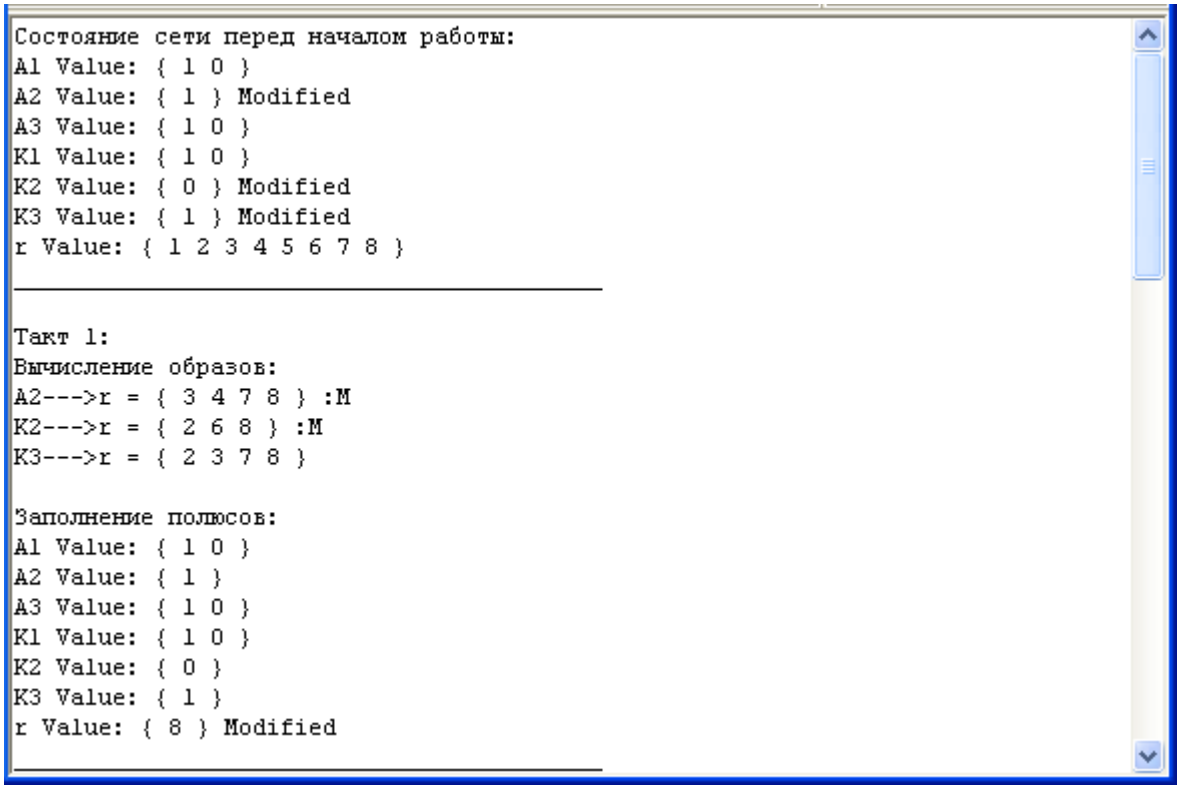


Рисунок 5.1 – Початок роботи моделі завдання логічного розпізнавання

Спираючись на цю подібність, можна визначити функції різних типів нейронних структур і описати в точних математичних і технічних термінах принципи функціонування мозку.

На рис. 5.2 зображений другий такт роботи моделі – останній ефективний такт. Хоча система виконає ще 2 такти, щоб перевірити умови стабільності мережі, значення змінних уже не міняються.

Рисунок 5.2 – Завершення роботи моделі завдання логічного розпізнавання

На рис. 5.2 можна бачити, що модель логічної мережі для вирішення завдання логічного розпізнавання завершила роботу.

Інтерпретація результатів в термінах вихідного завдання – при поставлених умовах завдання має єдине рішення. Усі три ознаки повинні бути присутнім. Крім зазначеної в умові інформації процес ДО1 обов'язково виконується. Таким чином, поставлене завдання вирішене в два такти.

У роботі створена програмна модель логічної мережі, яка надає можливості створення й налагодження моделей, проведення машинних експериментів і оцінки ефективності мережі. Її функціонування являє наочний приклад того, по якому принципу в майбутньому, можливо, будуть діяти ЕОМ, тому що логічні мережі – схемна реалізація формул, що описують алгебро-логічні структури, які у свою чергу є спеціальним математичним апаратом для формульного представлення відносин і дій над ними. Відносини інтерпретуються як думки

інтелекту, а дії над ними – як мислення. Кожний тип алгебро-логічних структур (а таких типів порівняно небагато) приводить до свого типу логічних мереж. При співставленні цих типів мереж з основними типами нейроструктур виявляється глибока подібність будови технічних і біологічних конструкцій.

У результаті опрацювання алгоритмів на програмній моделі зв'язки між двома різними визначеннями категорії з'ясувалося, що безоб'єктній категорії відповідає цілий клас ізоморфних категорій з об'єктами. Був ретельно опрацьований універсальний математичний апарат логічних мереж на основі алгебри скінченних предикатів, а точніше його центральний фрагмент, який ставиться до теорії про логічні простори – логічного аналізу.

У підсумку була знайдена інтерпретація категорії в термінах алгебри предикатів – предикатна категорія, причому для обох випадків: категорії з об'єктами й безоб'єктної категорії.

# ВИСНОВКИ

У результаті виконання атестаційної роботи магістра було здійснене рішення ряду завдань, що ставляться до питань взаємозв'язку між теорією категорій і алгеброю предикатів, побудовано елементи програмної моделі мозкоподібної ЕОМ. Зроблений аналіз предметної області, докладно розглянутий обраний математичний апарат. Розглянуті можливості його додатка до деяких завдань штучного інтелекту, зокрема, до вирішення проблем обробки інформації природною мовою, завданням логічного розпізнавання й комп'ютерного зору. Була побудована програмна модель – додаток, що реалізує математичні моделі, побудовані засобами алгебри предикатів з результатом у вигляді логічної мережі – графа, що представляє собою систему предикатних рівнянь у графічній формі, до рішення якої й зводяться поставлені завдання.

При вивченні теорії категорій поряд зі звичайним поняттям категорії зустрілося й більш загальне поняття безоб'єктної категорії. Був ретельно вивчений універсальний математичний апарат алгебри предикатів, розглянуте визначення логічної мережі. У підсумку була знайдена інтерпретація категорії в термінах алгебри предикатів – предикатна категорія, причому для обох випадків: категорії з об'єктами й безоб'єктної категорії.

Проведена формалізація досліджених моделей декларативною мовою, що описана у роботі, що й дозволяє будувати їх на ЕОМ за допомогою розробленої універсальної програмної моделі логічної мережі.

Наведено приклади рішення конкретних завдань із відповідних галузей на моделях, які застосовані в роботі, досліджений процес роботи моделей на ЕОМ і їх ефективність.

Зроблені висновки щодо можливостей наведеного підходу ефективно описувати й моделювати завдання з розглянутих областей.

# ПЕРЕЛІК ДЖЕРЕЛ ПОСИЛАННЯ

1. Про мозкоподобні ЕОМ / М.Ф. Бондаренко, З.В. Дудар, І.А. Єфімова, В.А. Лещинский, С. Ю. Шабанов-Кушнаренко // Радіоелектроніка й інформатика.–2014.–№ 2.–С. 21-38.

2. Un modèle d'indexation relationnel pour les graphes conceptuels fondée sur une interprétation logique: Phd thesis / Universitée Joseph Fourier. – Grenoble, 1998.-302 p.

3. Хьюбелл Д. Око, мозок, зір. – М.: Мир, 2010.–239 с.

4. Климушев В.Б. Моделювання кореляційного алгоритму розпізнавання образів на базі формальної граматики // Автоматизовані системи керування й прилади автоматики. –2001. – №115.– С. 54-74.

5. Шрейдер Ю. А., Шаров А. А. Системи й моделі. – М.: Радіо й зв'язок, 1982. – 152 с.

6. Плоткин Б.І. Універсальна алгебра, алгебраїчна логіка й бази даних. – М.: Мир, 1990.– 322 с.

7. Бондаренко М.Ф., Осика А.Ф. Автоматична обробка інформації природньою мовою. – Київ: УМК В, 1991. –143 с.

8. Алгебра предикатів і предикатних операцій./ М.Ф. Бондаренко, З.В. Дудар, Н.Т. Процай, В.В. Черкашин, В.А. Чикина, Ю.П. ШабановшКушнаренко // Радіоелектроніка й інформатика. – 2004.–№ 1. – С. 5-22.

9. Нильсон Н. Принципи штучного інтелекту. – М.: Радіо й зв'язок, 1985. – 373 с.

10. Шабанов-Кушнаренко Ю.П. Про теорію інтелекту // Проблеми біоніки. – Х.: Вища школа, 1979. – Вип. 22. – с. 3 – 11.

11. Шабанов-Кушнаренко Ю.П. Теорія інтелекту. Проблеми й перспективи. –Х.: Вища школа, 1987. –160 с.

12. Шабанов-Кушнаренко Ю.П. Теорія інтелекту. Математичні засоби. – Х.: Вища школа, 1984. –144 с.

13. Шабанов-Кушнаренко Ю.П. Теорія інтелекту. Технічні засоби. –Х.:

Вища школа, 1986. –136 с.

14. Голдблатт А, Топоси К. Категорний аналіз логіки. М.: Мир, 2013. 486 с.

15. Горелик А.Л., Скрипкін В.А. Методи розпізнавання. – М.: Вища школа, 1977. – 222 с.

16. C/C++. Програмування мовою високо рівня. – Спб.: Пітер, 2013. – 461 с.

17. Нечаєв В.І. Числові системи. М.: «Освіта», 2016. – 198 с.

18. Яценко В.В. (ред.) Введення в криптографію. – М., МЦНМО: "Черо", 2009, -349 с.

19. Боггс У., Боггс М. UML и Rational Rose. М.: ЛОРИ, 2014. 590 с.

20. Бондаренко М.Ф., Дударь З.В., Процай Н.Т., Черкашин В.В., Чикина В.А., Шабанов-Кушнаренко Ю.П. Алгебра предикатов и предикатных операций. // Радиоэлектроника и информатика. – 2004. № 3.

21. Бондаренко М.Ф., Дударь З.В., Ефимова И.А., Лещинский В.А., Шабанов-Кушнаренко С.Ю. О мозгоподобных ЭВМ. // Радиоэлектроника и информатика. – 2004. № 1.

22. О мозгоподобных ЭВМ / М.Ф. Бондаренко, З.В. Дударь, И.А. Ефимова, В.А. Лещинский, С. Ю. Шабанов-Кушнаренко // Радиоэлектроника и информатика.–2004.–№ 2.–С. 21-38.

23. Алгебра предикатов и предикатных операций./ М.Ф. Бондаренко, З.В. Дударь, Н.Т. Процай, В.В. Черкашин, В.А. Чикина, Ю.П. Шабанов-Кушнаренко // Радиоэлектроника и информатика. – 2004.–№ 1. – С. 5-22.

24. Мельникова Р.В. Алгебрологические модели морфологии и их применение в логических сетях. – Дисс. ... канд. техн. наук: 05.13.23. – Харьков, 2006. – 153 с.

25. Козяев Л.Л. Методы формализации и модели морфологических структур и их применение в системах искусственного интеллекта. – Дисс. ... канд. техн. наук: 05.13.23. – Харьков, 2006. – 151 с.