

НАЦІОНАЛЬНА АКАДЕМІЯ НАУК УКРАЇНИ
ІНСТИТУТ ТЕЛЕКОМУНІКАЦІЙ І ГЛОБАЛЬНОГО ІНФОРМАЦІЙНОГО
ПРОСТОРУ

Кваліфікаційна наукова
праця на правах рукопису

Ющенко Катерина Сергіївна

УДК 004.738.5; 004.021

ДИСЕРТАЦІЯ

**МОДЕЛІ ТА ІНСТРУМЕНТАЛЬНІ ЗАСОБИ ДЛЯ ПРИЙНЯТТЯ
РІШЕНЬ ЗА УМОВ НЕВИЗНАЧЕНОСТІ ПРИ
АВТОМАТИЗОВАНОМУ ВІДБОРІ ПЕРСОНАЛУ
НА ІНТЕРНЕТ-ПЛАТФОРМАХ**

122 – «Комп'ютерні науки»

Галузь знань 12 – Інформаційні технології

Подається на здобуття ступеня доктора філософії

Дисертація містить результати власних досліджень. Використання ідей,
результатів і текстів інших авторів мають посилання на відповідне джерело

К.С. Ющенко



Наукові керівники:
Гуляєв Кирило Дмитрович
кандидат технічних наук,
старший науковий співробітник.
Кряжич Ольга Олександрівна
кандидат технічних наук,
старший науковий співробітник.

Київ – 2023

АНОТАЦІЯ

Ющенко К.С. Моделі та інструментальні засоби для прийняття рішень за умов невизначеності при автоматизованому відборі персоналу на Інтернет-платформах. – Кваліфікаційна наукова праця на правах рукопису.

Дисертація на здобуття наукового ступеня доктора філософії за спеціальністю 122 «Комп'ютерні науки» – Інститут телекомунікацій і глобального інформаційного простору Національна академія наук України, Київ, 2023.

Дисертаційна робота присвячена розробці моделей та інструментальних засобів підтримки прийняття рішень за умов невизначеності в сучасних системах автоматизованого відбору персоналу на Інтернет-платформах.

У вступі обґрунтовано актуальність теми, розглянуто зв'язок роботи з науковими темами та актуальними напрямками, сформульовані мета та задачі дослідження, розкрито наукову новизну та практичну цінність.

У першому розділі проведений теоретичний аналіз джерел щодо розробки моделей та інструментальних засобів для прийняття рішень при автоматизованому відборі персоналу. Визначено, що при доволі широкому представленні робіт щодо підтримки рішень в сфері управління персоналом організації, в тому числі, з використанням сучасного інструментарію, роботи щодо застосування поєднання сучасних моделей, інструментальних засобів та Інтернет-технологій при роботі з персоналом залишилися на рівні теоретичних напрацювань. Проаналізовані методологічні підходи дозволили сформулювати поняття 3D резюме, яке пропонується розглядати через документ HTML-формату, який представляє професійні досягнення особи в трьох вимірах через текст, співбесіду з відеозаписом та тестування в режимі он-лайн. Визначені та обґрунтовані основні методи та підходи до реалізації задач роботи.

У другому розділі представлена реалізація підходу переходу та вибору, як дій, що часто повторюються, при тестуванні знань та вмінь у веб-сервісі з чітко означеними задачами. Основою до наукового пошуку стали абстрактні числові автомати, які вже знаходили свою реалізацію при відтворенні поведінки нервової системи живої істоти в розробці технологій штучного інтелекту. До уваги прийнято те, що абстрактний автомат функціонує в дискретному часі $t = 0, 1, 2, \dots$ і переходи між станами здійснюються миттєво. У кожен момент t дискретного часу автомат знаходиться в певному стані $a(t)$ з множини A станів автомату, причому в початковий момент часу $t = 0$ він завжди знаходиться в початковому стані a_0 . У момент часу t перебуваючи в стані $a(t)$, автомат здатний сприйняти на вхідному каналі сигнал $x(t) \in X$ і видати на вихідному каналі сигнал $y(t) = \lambda(a(t), x(t))$, переходячи в стан $a(t + 1) = \delta(a(t), x(t))$. Залежність вихідного сигналу від вхідного стану вказує про наявність пам'яті.

Подібне функціонування автомату дозволяє робити повтор дії та вибір наступного кроку (питання або завдання) в залежності від правильності попередньої дії (істини або хибності у вирішенні поставленого завдання). Зазначене може бути використане в розробці інструментального засобу з відбору персоналу для опису дій людини, що проходить тестування за допомогою мережі Інтернет, коли на початку є деякі вхідні слова (тезаурус спеціальності, що закладений у тести, завдання, співбесіду у дистанційному форматі), що описують розуміння респондентом теми співбесіди, а на виході, після складного перебору обмежень, є сформований системою висновок про відповідність даного претендента наведеним професійним вимогам. Звичайно, остаточне рішення приймає людина (менеджер по персоналу), але основний відбір здійснює саме система, яка разом з респондентом переходить за циклами вибору та запам'ятовує питання і відповіді на них, співвідносячи це з конкретною зареєстрованою в системі особою.

Використання абстрактних автоматів для реалізації дій повтору і вибору у алгоритмі моделювання інтелектуальної діяльності людини дозволяє

здійснювати переходи між питаннями в залежності від попередньої відповіді респондента. Наприклад, якщо відповідь претендента при заповненні 3D резюме не співпадає хоча б з одним з ключових слів, то наступне завдання 3D резюме не повинне активуватися.

В розділі реалізований підхід з програмування тригерів, як схеми зі стійкими станами перебування, допоки на зміниться керуючий вплив, і які дозволяють перетворити та запам'ятати інформацію, що отримала система, можливе за допомогою будь-яких сучасних мов програмування. Приклад програмування тригера буде наведений в практичному розділі роботи. Тип тригера, який може бути використаний при реалізації 3D резюме може бути обрано за способом організації логічних зв'язків, з лічильним входом, із затримкою, універсальні, комбіновані і, навіть, за способом запису інформації. Тип тригера обирається в залежності від задач 3D резюме.

Наведений евристичний алгоритм кодування станів мінімізує сумарне число змін елементів пам'яті на всіх переходах автомата. Це дозволяє виключити можливість вибору одного питання із множини питань для того ж самого користувача, а також вибору одного набору питань для різних користувачів, що проходять сегментацію чи навчання. Це виключає вплив людського фактору на автоматизовану систему навчання персоналу і дозволяє отримати об'єктивні результати.

У третьому розділі роботи розглядається задача розробки моделі для прийняття рішення при автоматизованому відборі персоналу шляхом тестування, коли в процесі перебору переходів від одного питання до іншого окрім задачі власне забезпечення цих переходів із запам'ятовуванням результатів входу і виходу, слід також реалізувати модель вибору завдань із множини / переліку всіх завдань, що закладені в систему.

При цьому слід врахувати, що завдання повинні поступати у порядку збільшення або зменшення складності для уточнення рівня підготовки працівника або кандидата на роботу. Завдання можуть бути з підказкою або без, із використанням тезаурусу професійного словника, забезпечувати деяку

ймовірність проходження завдань (відсоток виконаних завдань для отримання задовільного результату). Завдання не повинні повторюватися в межах виконання одного тестування і не повинні мати однакової черговості відкриття при низці перевірок знань працівників. В роботі для вирішення цієї наукової задачі було розвинуто підхід Г.П. Донця, В.А. Пепеляєва, О.М. Трофимчука про оптимальне розбиття на кількість груп та запропоновано стратегію щодо генерування вибору випадкового питання для формування множини питань в системі тестування знань працівника. Створено тестовий примірник програми-генератора та апробовано при формуванні вибірки з десяти випадкових завдань за номерами запису від 1 до 2000.

Розроблено модель з використанням алгоритму за оберненою функцією та за методом Неймана-Пірсона для генерації випадкового номера завдання за заданої аналітичної функції розподілу. Алгоритм перевірено при роботі з переліком з 2000 завдань.

Для того, щоб створити систему підбору кадрів з використанням засобів штучного інтелекту, необхідно, щоб система могла вирішувати задачу відповідності отриманої відповіді поставленому питанню. Тільки після цього, оцінюючи виконання попереднього завдання, для респондента може бути відкрите наступне завдання. Задачу відповідності відповіді можна вирішити за допомогою логіки висловлювань, зокрема, за допомогою обчислювання висловлювань або логічного програмування.

Словник обчислювань висловлювань може допомогти у формуванні висновку відносно нових висловлювань, отриманих від респондента. Правила побудови описують ті вирази, які є об'єктами мови, або формулами мови. Тобто, вислів є складеним висловлюванням, яке побудоване за певними мовними правилами.

Сукупність правил може бути представлена наступним чином:

а) базисом є слово або висловлювання, яке входить до тезаурусу сфери опитування і визнається формулою;

б) індукційним кроком є припущення того, що деякі ключові слова X та Y у відповіді респондента є формулами, у такому випадку формулами є деякі логічні висловлювання: кон'юнкція, диз'юнкція, логічне заперечення, імплікація, еквівалентність;

в) обмеженням правил є те, що застосовуються лише базис та індукційний крок для визначення істини або хибності.

Фактично застосування такого правила є рух по вузлах деякого дерева висловлювань: кореневий вузол – початкове слово, а далі вузли-зв'язки покрокові, які дозволяють на кожному індукційному кроці виявити відповідність відповіді поставленій задачі. У підсумку розглянутого питання наведено сукупність правил логіки, а також описано процес визначення відповідності відповіді поставленому питанню за допомогою дерева висловлювань. Представлено алгоритм знаходження відповідності отриманої відповіді за словами-еталонами та ключовим словом завдання.

Наведена модель процесу обчислення висловлювань з використанням базового словника, який сформовано за правилами логіки, що дозволяє з використанням засобів штучного інтелекту визначити, хибна чи істинна була відповідь. Для цього використано підхід, який у разі отримання неточної відповіді та невідповідності з висловлюванням Y_i , здійснює перехід до нового сформованого висловлювання Y_{i+1} , яке також не є абсолютно точним, але дозволяє співставити відповідь респондента з отриманням оцінки знань. При вірній відповіді та співпадінні з висловлюванням Y_{i+1} , процес завершується.

Зазначене враховує той аспект, що як і у випадку роботи з людиною, існує базовий словник, в який додаються необхідні терміни, на основі яких формуються логічні ланцюги з формули та індукційних кроків. Слова-еталони, які може використати респондент при відповіді на завдання, слід порівняти з ключовим словом, яке виступає базисом вірної відповіді на завдання. Базовий словник системи повинен забезпечувати дві вимоги: слова-еталони повинні бути упорядковані, а сам словник повинен мати інформаційну надлишковість. Виникнення нового слова відповіді, яке за індукційного кроку не суперечить

обмеженням, додається до базового словника. Реалізація цього питання дозволила розробити модель співвідношення точності отриманої відповіді для якісної оцінки знань.

Враховуючи те, що реалізація відбору персоналу, особливо на великих підприємствах чи рекрутингових агентствах, відбувається не поодібно, а з одночасною великою кількістю працівників, щодо яких відбувається тестування знань, задачею інформаційної технології роботи з персоналом є забезпечення багатопотокового ланцюга одночасного опитування персоналу. Наведено алгоритм здійснення переходів між згенерованими вибірками питань при одночасному тестуванні декількох респондентів за підходом, використовуваному у СМО. Враховано, що тестування відбувається найчастіше для груп осіб, а не поодиноких респондентів, тому є необхідність забезпечення багатопотокового ланцюга одночасного опитування персоналу. Зроблено перехід від моделі процесу обчислення висловлювань та якісної оцінки знань до забезпечення системою процесу такого перебору питань, який дозволяє формувати різні вибірки для кожного окремого респондента. Запропоновано для організації процесу здійснення переходів між згенерованими вибірками питань при одночасному тестуванні декількох респондентів використовувати підхід із застосуванням теорії систем масового обслуговування. Представлено алгоритм та лістинг програми.

Четвертий розділ роботи представляє експериментальну версію системи відбору персоналу у вигляді веб-реалізації 3D резюме. В процесі розробки 3D резюме був реалізований експеримент з використанням обчислювальної платформи для конструювання Arduino, який дозволив опрацювати механізм переходів між питаннями в залежності від правильної відповіді на попереднє питання. На цій основі було розроблене 3D резюме для тестування знань працівника (за приклад взяті вимоги до програміста C++).

У підсумку, на Інтернет-платформі було створене 3D резюме, опрацьований механізм перебору питань, переходів між окремими вкладками 3D резюме після виконання обов'язкової процедури підтвердження

відповідності знань вимогам наявної вакантної посади через аналіз ключових слів у письмовій відповіді респондента.

Початком роботи із 3D резюме є реєстрація в системі. За зареєстрованим логіном фіксується унікальна ознака об'єкта. За цією унікальною ознакою записи про дії користувача вносяться до бази даних. Кожен вхід до системи через спеціальну форму входу фіксується за датою, часом входу і виходу, виконаними або не вірно виконаними завданнями. Точка входу за часом в систему 3D резюме дозволяє відкрити початковий етап тестування – тестове письмове завдання. Такий підхід дозволяє за ключовими словами на самому початку складання 3D резюме визначити, наскільки людина володіє тезаурусом спеціальності та може в подальшому розібратися з іншими завданнями, необхідними для визначення компетентностей працівника. Такі ключові слова утримуються в базовому словнику системи 3D резюме. Безпосередньо інтерфейс роботи над тестовим завданням є класичним вікном для введення відповіді користувачем своєю рідною мовою чи, на необхідності, для написання програмного коду. Після того, як написання тестового завдання виконано, активуються кнопки двох інших вкладок 3D резюме – проходження тестів та відеопитання, для проведення опитування у відеорежимі.

Реалізовані процедури заміни номеру питання та тригерів у базі даних 3D резюме. Наведений опис та лістинг кодів реалізації таких процедур при управлінні базою даних 3D резюме.

Проведене тестування експериментальної версії 3D резюме. Для запобігання послідовному перебору всіх слів відповіді та пошуку відповідності в базовому словнику системи, що може займати багато часу, був розроблений ітераційний алгоритм визначення відповідності введеного слова ключовому слову з базового словника 3D резюме. Аналіз результатів експерименту та математичного моделювання за технологією GOMS щодо тестування експериментальної версії 3D резюме дозволив визначити значення дисперсії розподілу сеансних результатів для сигнатур, які характеризуються

найбільшим розкидом значень. Був отриманий розподіл близький до розподілу Стюдента. Довірча ймовірність складає $P = 95\%$.

Результати проведеної роботи впроваджені у діяльність Товариства з обмеженою відповідальністю «Будівельно-проектна фірма Україна» для створення форми навчання кадрів з техніки безпеки при виконанні будівельних робіт та в роботу «Академії праці, соціальних відносин і туризму» для використання у навчальному процесі.

Ключові слова: автомат Мура, перехід, логіко-ймовірнісних підхід, сукупність, імітаційна модель, математична модель, 3D резюме, GOMS, лістинг.

ABSTRACT

Iushchenko K. S. Models and tools for making decisions in automated personnel selection on Internet platforms in conditions of uncertainty. – Qualifying scientific work on manuscript rights.

Dissertation for the degree of Doctor of Philosophy in specialty 122 "Computer Science" – Institute of Telecommunications and Global Information Space, National Academy of Sciences of Ukraine, Kyiv, 2023.

The dissertation work is devoted to the development of models and tools for supporting decision-making under conditions of uncertainty in modern systems of automated personnel selection on Internet platforms.

Conducted theoretical analysis of sources regarding the development of a system of selection of personnel. It was determined that there is an open question regarding the combination of models and tools for personnel selection with modern web technologies. The analyzed methodological approaches made it possible to formulate the concept of a 3D resume, which is proposed to be considered through an HTML-format document that represents a person's professional achievements in three dimensions through text, video interview and online testing. The main methods and approaches to the implementation of work assignments are defined and substantiated.

The implementation of the approach of interaction of information and communication technologies in a web service with clearly defined tasks is presented. Abstract numerical automata became the basis for scientific research, which had already found their implementation in the reproduction of the behavior of the nervous system of a living being in the development of artificial intelligence technologies. It is taken into account that the abstract automaton functions in a discrete time $t = 0, 1, 2, \dots$ and transitions between states are carried out instantaneously. At each moment t of discrete time, the automaton is in a certain state at from the set A of states of the automaton, and at the initial moment of time

$t=0$ it is always in the initial state a_0 . At the moment of time t , being in the state $a(t)$, the automaton is able to receive the signal $x(t) \in X$ on the input channel and issue the signal $y(t) = \lambda(a(t), x(t))$ on the output channel, passing to the state $a(t+1) = \delta(a(t), x(t))$. The dependence of the output signal on the input state indicates the presence of memory.

Such functioning of the automaton allows you to repeat the action and choose the next step (question or task) depending on the correctness of the previous action (truth or error in solving the task). This can be used in the development of artificial intelligence to describe the actions of a person undergoing testing using the Internet, when at the beginning there are some input words (thesaurus of the specialty embedded in the tests, assignments, interview in a remote format) that describe the respondent's understanding of the topic interview, and at the output, after a complex selection of restrictions, there is a conclusion formed by the system about the compliance of this applicant with the specified professional requirements. Of course, the final decision is made by a person (personnel manager), but the main selection is carried out by the system itself, which together with the respondent "learns" by going through selection cycles and remembers questions and answers to them, correlating it with a specific person registered in the system.

The use of abstract automata for the implementation of repetition and selection actions in the algorithm for modeling human intellectual activity allows transitions between questions depending on the respondent's previous answer. For example, if the applicant's answer when filling out a 3D resume does not match at least one of the keywords, then the next task of the 3D resume should not be activated.

The section implements the approach of programming triggers, as schemes with stable states of stay until the controlling influence changes, and which allow to transform and remember the information received by the system, possible with the help of any modern programming languages. An example of trigger programming will be given in the practical section of the work. The type of trigger that can be used in the implementation of 3D summary can be chosen according to the method of

organizing logical connections, with a counting input, with a delay, universal, combined and, even, according to the method of recording information. The type of trigger is chosen depending on the tasks of the 3D resume.

The given heuristic algorithm for encoding states minimizes the total number of changes of memory elements at all transitions of the automaton. This eliminates the possibility of selecting one question from multiple questions for the same user, as well as selecting the same set of questions for different users undergoing segmentation or training. This excludes the influence of the human factor on the automated staff training system and allows obtaining objective results.

The task of developing an information system using elements of artificial intelligence is considered, when in the process of sorting out transitions from one question to another in the system of selection, segmentation and training of personnel, in addition to the task of providing these transitions with memorizing the results of entry and exit, a model of task selection should also be implemented from the set / list of all tasks included in the system.

An algorithm based on the inverse function and the Neumann-Pearson method was developed to generate a random task number based on a given analytical distribution function. The algorithm was tested when working with a list of 2000 tasks.

In order to create a system of personnel selection, segmentation and training using artificial intelligence tools, it is necessary for the system to be able to solve the problem of matching the received answer to the question. Only after that, evaluating the performance of the previous task, the next task can be opened for the respondent. The answer matching problem can be solved using statement logic, in particular, statement computation or logic programming.

An utterance count dictionary can assist in inferring new utterances received from a respondent. Construction rules describe those expressions that are language objects, or language formulas. That is, an expression is a compound statement that is built according to certain linguistic rules. In fact, the application of such a rule is movement along the nodes of a tree of statements: the root node is the initial word,

and then there are step-by-step connection nodes that allow at each induction step to find out the correspondence of the answer to the given task. As a result of the considered question, a set of rules of logic is given, and the process of determining the appropriateness of the answer to the question using the tree of statements is also described. The algorithm for finding the correspondence of the received answer based on reference words and the keyword of the task is presented.

A model of the process of calculating statements using a basic dictionary, which is formed according to the rules of logic, is presented, which allows using artificial intelligence to determine whether the answer was true or false. For this, an approach was used, which in case of receiving an inaccurate answer and inconsistency with the statement Y_i , makes a transition to the newly formed statement Y_{i+1} , which is also not absolutely accurate, but allows to compare the respondent's answer with obtaining a knowledge assessment. If the answer is correct and matches the statement Y_{i+1} , the process ends.

This takes into account the aspect that when using artificial intelligence, just as in the case of working with a person, there is a basic dictionary to which the necessary terms are added, on the basis of which logical chains of formulas and induction steps are formed. Standard words that the respondent can use when answering the task should be compared with the key word that serves as the basis for a correct answer to the task. The base dictionary of the system must meet two requirements: the reference words must be ordered, and the dictionary itself must have information redundancy. The occurrence of a new answer word that does not contradict the constraints during the induction step is added to the base dictionary. The implementation of this question made it possible to develop a model of the ratio of the accuracy of the received answer for the qualitative assessment of knowledge.

Taking into account the fact that the implementation of recruitment, segmentation and training of personnel, especially in large enterprises or recruiting agencies, does not take place individually, but with a large number of employees at the same time, in respect of whom knowledge testing takes place, the task of information technology for working with personnel is to provide a multi-threaded

chain of simultaneous personnel surveys. An algorithm for making transitions between generated question samples during simultaneous testing of several respondents according to the approach used in mass service systems is given.

The work presents an experimental version of the personnel selection, segmentation and training system using elements of artificial intelligence in the form of a web implementation of a 3D resume. In the process of developing a 3D resume, an experiment was carried out using a computing platform for the design of Arduino, which allowed to work out the mechanism of transitions between questions depending on the correct answer to the previous question. On this basis, a 3D resume was developed for testing the employee's knowledge (the requirements for a C++ programmer were taken as an example).

As a result, a 3D resume was created on the web platform, a mechanism for sorting through questions, transitions between separate tabs of the 3D resume was developed after the mandatory procedure for confirming knowledge compliance with the requirements of the available vacant position through the analysis of keywords in the written answer of the respondent.

Implemented procedures for replacing the question number and triggers in the 3D resume database. The description and listing of the codes for the implementation of such procedures in the management of the 3D resume database is given.

The experimental version of the 3D resume has been tested. In order to prevent the sequential search of all answer words and the search for a match in the base dictionary of the system, which can take a lot of time, an iterative algorithm was developed to determine the correspondence of the entered word to the key word from the base dictionary of 3D resume. The analysis of the results of the experiment and mathematical modeling using the GOMS technology for testing the experimental version of the 3D resume made it possible to determine the value of the dispersion of the distribution of session results for the signatures characterized by the largest spread of values. A distribution close to the Student's distribution was obtained. Confidence probability is $P = 95\%$.

The results of the work are implemented in the activities of the Limited Liability Company "Construction and Design Firm Ukraine" to create a form of training for personnel on safety techniques when performing construction works and in the work of the "Academy of Labor, Social Relations and Tourism" for use in the educational process.

Keywords: Moore's automaton, transition, logical-probabilistic approach, set, simulation model, mathematical model, 3D summary, GOMS, listing.

СПИСОК ОПУБЛІКОВАНИХ ПРАЦЬ ЗА ТЕМОЮ ДИСЕРТАЦІЇ

Наукові праці, в яких опубліковані основні наукові результати дисертації

Публікації у фахових виданнях:

1. Ющенко К.С. Підхід до автоматизації процесу підбору кадрів за допомогою 3D резюме. Математичні машини і системи. 2022. № 2. С. 29–39. **(кат. Б)**

2. Кряжич О.О., Ющенко К.С., Іцкович В.Є., Купрін О.М. Особливості алгоритмізації процесів мінімізації похибок апроксимації при вирішенні прикладних задач. Математичні машини і системи. – 2023 – №1 – сс. 118 – 129. **(кат. Б)**

3. Kryazhych O., Itskovych V., Iushchenko K., Hrytsyshyna V., Bruvier D., Nykytyuk V., Vodnarchuk I. (2023) The use of abstract moore automaton to control the sensors of a service-oriented alarm and emergency notification network. Scientific Journal of TNTU (Tern.), vol 109, no 1, pp. 111–120. **(кат. Б)**

Публікації у виданнях, що входять до міжнародних науково-метричних баз:

4. Kryazhych, O., Itskovych, V., Iushchenko, K., Kuprin, O. (2023). Features in solving individual tasks to develop service-oriented networks using dynamic programming. Eastern-European Journal of Enterprise Technologies, 1 (4 (121)), 34–40. doi: <https://doi.org/10.15587/1729-4061.2023>. **(Scopus, Q3)**

5. Kryazhych, O., Kovalenko, O., Itskovych, V., & Iushchenko, K. (2023). The modeling of changes in the specific activity of tritium in plants. EUREKA: Physics and Engineering, (3), 3-14. <https://doi.org/10.21303/2461-4262.2023.002877> **(Scopus, Q2)**

6. Гуляєв, К. Д., Ющенко, К. С., Купрін, О. М. (2023). Застосування абстрактних автоматів Мілі та Мура для реалізації алгоритмів рекомендації та вибору. International Scientific Technical Journal "Problems of Control and Informatics", 67(6), с. 14–24. doi: <https://doi.org/10.34229/1028-0979-2022-6-2>.

(кат. А)

Наукові праці, які засвідчують апробацію матеріалів дисертації:

7. Ющенко К.С. Засоби штучного інтелекту у системах підготовки персоналу // Історія, сучасний стан та тенденції цифрового розвитку суспільства. / Матеріали 10-ої Міжнар. наук.-практ. конф. «Глушковські читання», Київ, 2021 р. / Уклад.: Р.М. Богачев, В.Д. Піхорович, А.Ю. Самарський, М.І. Сторожик. – Київ, 2021. – с. 194 – 196.

8. Ющенко К. С. Цифровізація процесів роботи з персоналом / Інформаційно-комунікаційні технології та сталий розвиток // Колективна монографія за матеріалами XXI Міжнародної науково-практичної конференції (Київ, 14-16 листопада 2022 р.) / За заг. ред. С.О. Довгого. – К.: ТОВ «Видавництво «Юстон», 2022. – сс. 206 – 207. ISBN 978-617-7854-76-9.

ЗМІСТ

| | |
|---|----|
| ЗМІСТ..... | 18 |
| ПЕРЕЛІК УМОВНИХ ПОЗНАЧЕНЬ..... | 22 |
| ВСТУП..... | 23 |
| РОЗДІЛ 1. МОДЕЛІ ТА ІНСТРУМЕНТИ ПІДТРИМКИ ПРИЙНЯТТЯ РІШЕНЬ ПРИ АВТОМАТИЗОВАНОМУ ВІДБОРІ ПЕРСОНАЛУ З ВРАХУВАННЯМ СУЧАСНИХ ВИМОГ КАДРОВОГО МЕНЕДЖМЕНТУ | 30 |
| 1.1 Сучасні парадигми кадрового менеджменту в автоматизації процесів відбору кадрів..... | 30 |
| 1.2 Керуючі впливи, обмеження та вхідні параметри при розробці моделей та інструментальних засобів для прийняття рішень з відбору кадрів за умов невизначеності..... | 38 |
| 1.3 Використання інструментів штучного інтелекту в сучасних системах прийняття рішень з відбору кадрів..... | 45 |
| 1.4 Технологія дослідження та методи вирішення поставлених задач..... | 52 |
| 1.4.1 Технологія дослідження..... | 52 |
| 1.4.2 Методи реалізації роботи..... | 53 |
| 1.5 Висновки за розділом 1..... | 57 |
| РОЗДІЛ 2. РЕАЛІЗАЦІЯ ДІЙ ПОВТОРУ ТА ВИБОРУ ДЛЯ РЕАЛІЗАЦІЇ ІНСТРУМЕНТАЛЬНОГО ЗАСОБУ З ВІДТВОРЕННЯ ІНТЕЛЕКТУАЛЬНОЇ ДІЯЛЬНОСТІ ЛЮДИНИ..... | 59 |

| | | |
|--|---|-----|
| 2.1 | Використання абстрактного автомату для створення механізму відтворення людської поведінки..... | 59 |
| 2.2 | Мінімізація числа внутрішніх станів автомату щодо задачі створення 3D резюме..... | 64 |
| 2.3 | Евристичний алгоритм кодування станів автомату для 3D резюме..... | 70 |
| 2.4 | Формування функцій перемикавання та функцій виходу автомата..... | 76 |
| 2.5 | Побудова функціональної схеми автомата..... | 81 |
| 2.6 | Висновки за розділом 2..... | 87 |
| РОЗДІЛ 3. ЛОГІКО-ЙМОВІРНІСНИЙ ПІДХІД ДО ФОРМУВАННЯ МНОЖИНИ ЗАВДАНЬ ДЛЯ НЕЗАЛЕЖНИХ ВИПРОБУВАНЬ В СИСТЕМАХ ПІДБОРУ КАДРІВ..... | | 89 |
| 3.1 | Генерування вибору випадкового питання для формування множини питань в системі тестування знань працівника..... | 89 |
| 3.2 | Генерація випадкового номера завдання за заданої аналітичної функції розподілу..... | 93 |
| 3.3 | Логічний підхід до вирішення задачі відповідності отриманої відповіді поставленому питанню..... | 98 |
| 3.3.1 | Синтаксис обчислення висловлювань..... | 98 |
| 3.3.2 | Алгоритм знаходження відповідності отриманої відповіді за словами-еталонами та ключовим словом завдання..... | 100 |

| | | |
|--|---|-----|
| 3.4 | Використання правил логіки при формуванні базового словника системи тестування знань і навичок персоналу..... | 102 |
| 3.4.1 | Модель процесу обчислення висловлювань..... | 102 |
| 3.4.2 | Модель співвідношення точності отриманої відповіді..... | 106 |
| 3.5 | Алгоритм здійснення переходів між згенерованими вибірками питань при одночасному тестуванні декількох респондентів..... | 108 |
| 3.6 | Висновки за розділом 3..... | 117 |
| РОЗДІЛ 4. РОЗРОБКА ІНСТРУМЕНТАЛЬНОГО ЗАСОБУ ДЛЯ ПРИЙНЯТТЯ РІШЕНЬ ЗА УМОВ НЕВИЗНАЧЕНОСТІ ПРИ ВІДБОРІ КАДРІВ ЧЕРЕЗ ІНТЕРНЕТ-ПЛАТФОРМУ..... | | 120 |
| 4.1 | Реалізація механізму відтворення людської поведінки при підборі кадрів за умов невизначеності..... | 120 |
| 4.2 | Реалізація процедур заміни номеру питання та тригерів у базі даних 3D резюме..... | 128 |
| 4.3 | Тестування експериментальної версії 3D резюме..... | 133 |
| 4.3.1 | Імітаційна модель інтерфейсу 3D резюме з тестуванням користувача за ключовим словом..... | 133 |
| 4.3.2 | Реалізація підходу прискорення обробки введених слів відповіді..... | 136 |
| 4.3.3 | Результати математичного моделювання та експериментального тестування робочої версії 3D резюме | 138 |
| 4.4 | Висновки за розділом 4..... | 139 |

| | |
|---|-----|
| ВИСНОВКИ..... | 141 |
| СПИСОК ВИКОРИСТАНИХ ДЖЕРЕЛ..... | 143 |
| ДОДАТКИ..... | 156 |
| Додаток А. Акти впровадження результатів дисертаційної роботи..... | 157 |
| Додаток Б. Алгоритм виконання розрахунків матриці переходів автомату М для застосування евристичного методу кодування станів..... | 159 |
| Додаток В. Приклад логічного парадоксу при вирішенні задачі щодо двоканальної СМО з необмеженою чергою..... | 173 |
| Додаток Г. Реалізація комп'ютерного експерименту з використання абстрактного автомату для створення системи імітації людської поведінки в процесі навчання..... | 175 |
| Додаток Д. Лістинг коду реалізації 3D резюме..... | 180 |
| Додаток Ж. Приклад реалізації алгоритму виконання процедури обчислення відповідності ключовому слову | 203 |
| Додаток К. Данні математичного і імітаційного моделювання.... | 205 |

ПЕРЕЛІК УМОВНИХ ПОЗНАЧЕНЬ

- 3D – те, що має три виміри.
- ETL – витяг, перетворення та завантаження (інструментарій).
- GOMS – модель людського інформаційного процесора спостереження за людиною і комп'ютером (Goals, Operators, Methods, Selection Rules).
- GUI – графічні елементи інтерфейсу (Graphical User Interface).
- KLM – модель рівня натискання клавіш (Keystroke-Level Model).
- MySQL – система керування реляційними базами даних.
- NUI – природній інтерфейс користувача (Natural User Interface).
- RUI – елементи інтерфейсу.
- TUI – текстовий блок інтерфейсу (Text User Interface).
- VUI – голосові команди (Voice User Interface).
- WIMP – платформа вікон, значків, меню та вказівок (Windows, Icons, Menus, Pointing Device).
- БД – база даних.
- ІС – інформаційні системи.
- СМО – система масового обслуговування.
- ШІ – штучний інтелект (англ. – Artificial intelligence, AI).

ВСТУП

При вирішенні питань, що стосуються відносин «людина – комп'ютер», особливо тоді, коли за допомогою різноманітних інструментальних засобів відбувається прийняття рішень стосовно кадрів організації, завжди актуальним залишається питання, яким чином наблизити «машинний інтелект» до людської поведінки. В різноманітних роботах з гуманітарних напрямків досліджень, таких, як психологія, ергономіка, соціологія, менеджмент, філософія, останнім часом піднімаються питання стосовно зв'язку «людина – комп'ютер» щодо розподілу функцій обробки інформації, прийняття рішень без участі людини, дослідження складних процесів, надання рекомендацій на основі алгоритмів, обробки великих даних тощо. При цьому найбільші дискусії точаться щодо алгоритмізації процесів вибору та прийняття рішення, при яких людина виводиться з контуру управління, а її місце займається сучасними інформаційними технологіями. У суперечках обговорюються переваги людини в ланцюжку «людина – комп'ютер», які полягають у можливості долати невизначеності нестандартними методами, в залежності від ситуації та керуючих впливів, гнучко формулювати цілі та задачі управління. Проте, ніхто не заперечує, що людина, на відміну від комп'ютера, може мати суб'єктивний погляд на проблему, керуватися емоціями чи мати обмеження через затримку у часі при прийнятті складного обґрунтованого рішення. Зазначене цілком характерне для такої сфери діяльності, як управління персоналом. Враховуючи, що вимоги кадрового менеджменту у сучасних організаціях постійно змінюються, кадри стають не просто людськими ресурсами, а капіталом, потенціалом організації, яка існує в середовищі, де на ринку праці йде боротьба за кращих. Тому в зазначених умовах моделі та інструментальні засоби для прийняття рішень за умов невизначеності при автоматизованому відборі персоналу, особливо – на основі Інтернет-технологій, стають затребуваним інструментом кадрового

менеджменту. Від автоматизованих технологій відбору персоналу залежить майбутній розвиток будь-якої організації, її виживання у конкурентній боротьбі сучасного високотехнологічного ринку, що вказує на необхідність розробки нових моделей та інструментальних засобів для прийняття рішень за умов невизначеності при автоматизованому відборі персоналу на Інтернет-платформах та підкреслює **актуальність теми дослідження**.

Розробка моделей та інструментальних засобів для прийняття рішень при автоматизованому відборі персоналу має широке теоретичне підґрунтя. Так, роботи R. Duchessi, D. O'Keefe, V. Kolbjornsrud, M. El Khatib, O'Leary, H. McGrath, J. Ndiaye, B. M. Mohsen, O. Брінцевої, А. Матвійчука присвячено новим підходам, методам і моделям при алгоритмізації процесів роботи з персоналом. Серед українських дослідників далеко за межами України відомі роботи щодо управління персоналом з використанням елементів штучного інтелекту доктора технічних наук А.І. Шевченка. Безпосередньо технологія дослідження роботи базується на математичних підходах, закладених науковою школою О.М. Трофимчука, на роботах і розробках співробітників Інституту телекомунікацій і глобального інформаційного простору НАН України – С.О. Довгого, К.Д. Гуляєва, О.В. Копійки.

Не зважаючи на багаторічну розробку автоматизованих засобів з відбору персоналу можна зазначити, що поєднання різних інструментальних засобів та моделей для втілення систем підтримки прийняття рішень для відбору персоналу через Інтернет-платформи залишаються недостатньо розробленою. Як правило, інструментальні засоби для роботи з персоналом в мережі Інтернет представлені тестами та застосуванням комунікаційних технологій, де машині відведена роль статичного засобу для візуалізації алгоритму перебору питань, а людина – менеджер з персоналу, – виступає активним елементом, який приймає, обробляє, аналізує отриману інформацію (відповіді респондента) та приймає остаточне рішення. Тож, все перелічене, в сукупності, визначає **актуальність завдань дисертаційних досліджень**.

Зв'язок роботи з науковими програмами, планами, темами. Наукова спрямованість дисертації відповідає напрямкам науково-технічної політики України, Закону України «Про наукову і науково-технічну діяльність» (в останній редакції). Окремі дослідження виконувалися в рамках науково-дослідних Інституту телекомунікацій і глобального інформаційного простору НАН України, зокрема: «Створення когнітивних засобів забезпечення процесів проведення інформаційно-аналітичного аналізу науково-технічних досягнень» (№ ДР 0118U006274) та «Розробка елементів штучного інтелекту в задачах математичного прогнозування, робототехніки та адитивних технологій» (№ ДР 0121U110827).

Мета й завдання дослідження. Метою роботи є розробка моделей та інструментальних засобів для прийняття рішень за умов невизначеності при автоматизованому відборі персоналу на Інтернет-платформах.

Для досягнення мети дослідження було поставлено такі завдання:

– *узагальнити й систематизувати* існуючі моделі, методи та підходи при реалізації систем відбору персоналу, на цій основі сформувані базис дослідження та вирішення поставлених задач;

– на базі теорії абстрактних автоматів *запропонувати та обґрунтувати* створення інструментів повтору та вибору для алгоритмізації інтелектуальної діяльності людини при вирішенні логічних завдань або практичних задач, які вирішуватимуть через алгоритм переходу вибір між завданнями за рівнем складності в залежності від результатів відповіді на попередні завдання із запам'ятовуванням отриманого результату;

– *обґрунтувати та розробити* логіко-ймовірнісну модель формування множини завдань з відповідними алгоритмами реалізації для проведення незалежного оцінювання знань, вмінь та навичок в системах підбору персоналу, що забезпечуватимуть унікальність вибору завдань для кожного респондента та здійснювати перевірку відповідності відповіді поставленому питанню з розділенням за критерієм точності;

– на базі розроблених моделей та алгоритмів *реалізувати інструмент та провести його експериментальне діагностування* у вигляді 3D резюме на Інтернет-платформі з реалізацією моделі логічного вибору питання у разі попередньої помилки респондента та запам'ятовування результату.

Об'єкт дослідження – процеси автоматизації підтримки прийняття рішень за умов невизначеності при відборі персоналу за допомогою Інтернет-технологій.

Предмет дослідження – моделі та інструментальні засоби підтримки прийняття рішень при відборі персоналу за умов невизначеності на Інтернет-платформах.

Методи дослідження. Використовувався комплекс загальнонаукових та статистичних методів, математичного моделювання, прикладного програмування, теорії абстрактних автоматів, математичної логіки, теорії множин та інш.

Наукова новизна одержаних результатів міститься в наукових положеннях, що виносяться на захист та у яких:

1. *Уперше* розроблений інструментальний засіб, що відтворює дії людини, яка проходить тестування за допомогою мережі Інтернет, де на початку є деякі вхідні слова (тезаурус спеціальності, що закладений у тести, завдання, співбесіду у дистанційному форматі), які описують розуміння респондентом теми співбесіди, а на виході, після складного перебору обмежень, є сформований системою висновок про відповідність даного претендента наведеним професійним вимогам;

2. *Уперше* виконана пілотна реалізація 3D резюме на веб-платформі з експериментальним тестуванням робочої версії, визначенням терміну 3D резюме та розробкою методології створення;

3. *Дістали подальшого розвитку* підхід Г.П. Донця, В.А. Пепеляєва, О.М. Трофимчука про оптимальне розбиття на кількість груп, яка розширена шляхом алгоритмізації моделі щодо генерування вибору випадкового питання для формування множини питань в системі тестування знань працівника, а

також здійснюється логічний підхід із застосування синтаксису обчислювання висловлювань для вирішення задач відповідності отриманої відповіді поставленому питанню або для здійснення переходів між згенерованими вибірками питань при одночасному тестуванні декількох респондентів;

4. *Дістали подальшого розвитку моделі* за оберненою функцією та за методом Неймана-Пірсона для генерації випадкового номера завдання за заданої аналітичної функції розподілу, які, на відміну від існуючих дозволяють знаходити відповідності отриманої відповіді за словами-еталонами або за ключовим словом завдання, а також дозволяють реалізовувати систематизацію вибірки з базового словника системи з визначенням відповідностей.

Достовірність та обґрунтованість результатів. Результати роботи є достовірними, оскільки їх було отримано в ході математичного та чисельного моделювання, програмування, а також проведення досліджень із застосуванням апробованих експериментальних методів та стандартних методів статистичної обробки.

Достовірність основних положень та результатів дисертації доведено:

- використанням апробованих методів математичного та чисельного аналізу, статистичного спостереження та експерименту;
- використанням сертифікованих комп'ютерних програм та інтегрованих середовищ розробки;
- достатньою відповідністю результатів математичного моделювання та експериментальних досліджень.

Наукові положення, висновки та рекомендації **обґрунтовані**, тому що базуються на теоремах і методах математичного та чисельного аналізу, законах математичної логіки та спостереженнях математичної статистики. Розроблені алгоритми пройшли тестування на основі статистичних даних, робота з інтерфейсом програми протестована за допомогою користувачів.

Практичне значення отриманих результатів. Особливістю вирішених задач є те, що запропонований підхід може бути використаний в розробці

систем управління персоналом із використанням засобів штучного інтелекту як через прикладні додатки на персональних комп'ютерах, що було наведено в роботі, так і реалізованих на Інтернет-платформах. Розроблена теоретико-методологічна база, нові моделі та алгоритми дозволять побудувати новий клас систем управління персоналом, покращити процеси відбору та сегментації кадрів, проводити об'єктивне оцінювання знань та навичок працівників.

Реалізація роботи. Результати досліджень впроваджено:

1. В роботу Товариства з обмеженою відповідальністю «Будівельно-проектна фірма Україна» для створення форми навчання кадрів з техніки безпеки при виконанні будівельних робіт

2. В діяльність «Академії праці, соціальних відносин і туризму» для використання у навчальному процесі.

Документи, які підтверджують впровадження результатів досліджень, наведені в Додатку А.

Особистий внесок автора в роботи, опубліковані в співавторстві:

Автором самостійно отримані головні результати дисертаційного дослідження. В опублікованих у співавторстві наукових працях здобувачем здійснено (згідно «Списку опублікованих праць за темою дисертації»): у роботі [2] – вирішення прикладної задачі за тематикою дослідження, математичне моделювання; у статті [3] – проведення теоретичного аналізу за темою дослідження; у роботі [4] – математичне та імітаційне моделювання; у роботі [5] – вивчення особливостей алгоритмізації підходу тестування персоналу, що має специфічні знання та вміння, апробація 3D резюме; у статті [6] – побудова алгоритмів та розрахунки до абстрактного автомату Мура.

Апробація результатів дисертації. Результати дисертації обговорювалися та доповідалися на двох міжнародних науково-практичних конференціях: XXI Міжнародній науково-практичній конференції «Інформаційно-комунікаційні технології та сталий розвиток» (Київ, 14-16 листопада 2022 р.), X Міжнародній науково-практичній конференції

«Глушковські читання: Історія, сучасний стан та тенденції цифрового розвитку суспільства» (Київ, 2021 р.).

У повному обсязі дисертація доповідалася у 2023 р. на розширеному науково-технічному семінарі Інституту телекомунікацій та глобального інформаційного простору Національної Академії наук України під керівництвом доктора технічних наук, професора, члена-кореспондента НАН України Трофимчука О.М.

Публікації. Основні наукові результати дисертаційної роботи опубліковані у 8 публікаціях у професійних фахових виданнях: дві з них – статті, що проіндексовані в наукометричній базі SCOPUS; одна з них – стаття, що відноситься до журналу, який входить до категорії А у затверджених МОН України виданнях. Загалом п'ять статей – у затверджених МОН України виданнях. Серед публікацій, які додатково відображають наукові результати дисертації, 2 надрукованих доповіді у матеріалах міжнародних наукових та науково-практичних конференцій.

Структура й обсяг дисертації. Дисертаційна робота містить вступ, чотири розділи, висновки, додатки, список використаних джерел. Загальний обсяг дисертації – 205 сторінок, обсяг основного тексту – 122 сторінки. Робота містить 23 таблиці, 30 рисунків, додатки на 50 сторінках. Список використаних джерел складається зі 112 найменувань.

РОЗДІЛ 1

МОДЕЛІ ТА ІНСТРУМЕНТИ ПІДТРИМКИ ПРИЙНЯТТЯ РІШЕНЬ ПРИ АВТОМАТИЗОВАНОМУ ВІДБОРІ ПЕРСОНАЛУ З ВРАХУВАННЯМ СУЧАСНИХ ВИМОГ КАДРОВОГО МЕНЕДЖМЕНТУ

1.1 Сучасні парадигми кадрового менеджменту в автоматизації процесів відбору кадрів

В основі будь-якої інформаційної технології покладено саме дані, а не їх носій [1]. Проте лише носій інформації висуває вимоги, які закладаються в основи створення автоматизованих систем управління персоналом: методи обробки, використання математичних та статистичних підходів до аналізу даних та моделювання рішення [2].

У ХХ сторіччі в кадровому менеджменті були сформовані дві основних доктрини роботи з персоналом: наукова організація праці та людські відносини [3]. Якщо перша була зорієнтована на оптимізації організаційних, технічних і соціальних компонентів, де всі процеси могли бути описані за допомогою алгоритмів, то друга доктрина намагалася використовувати морально-психологічні та соціально організаційні фактори для досягнення певного економічного ефекту в організації і важко піддавалася алгоритмізації. І коли з'явилися перші алгоритми штучного інтелекту (ШІ, або англ. – Artificial intelligence, AI) у 1960, то однією із задач використання нової технології були саме задачі управління персоналом підприємств [4]. Однак основний розвиток ШІ в роботі з кадрами отримав на початку ХХІ століття [5], і саме тоді були висунуті задачі щодо автоматизації процесів управління персоналом в межах чотирьох основних парадигм у кадровому менеджменті:

– доктрина наукової організації праці;

- доктрина людських стосунків;
- доктрина контракції індивідуальної відповідальності;
- доктрина командного менеджменту [6].

Зазначені парадигми підштовхнули до розробки інструментів управління розвитком кадрового потенціалу підприємств за допомогою штучного інтелекту. За роботою [7] можна навести аналіз сучасних інструментальних засобів для прийняття рішень з управління персоналом організації, в тому числі – ті, що застосовуються при відборі персоналу (табл. 1).

Таблиця 1.1

Аналіз інструментальних засобів для прийняття рішень
з управління персоналом організації

| № | Процеси | Інструментальні засоби | Коротка характеристика | Переваги | Недоліки |
|---|-----------------------|---|---|---|---|
| 1 | 2 | 3 | 4 | 5 | 6 |
| 1 | Добір персоналу | Resume Matcher, Skillz, TalentTech Sever.AI, Hurma System | Онлайн тестування навичок та компетенцій, систематизація інформації, вибір працівника за заданими параметрами | Мутьти-платформеність, хмарні технології | Складність використання в Україні через недостатню цифровізацію населення та зависока ціна |
| 2 | Сегментація персоналу | Veriato 360, Workday, Yva.ai 3.0, Isaak Status Today by Glickon | Слідкування за роботою працівників, моніторинг даних про знання та навички | Більшість працює через сайт, мультиплатформеність | Не всі параметри підходять для системи моніторингу, завантаження пам'яті пристроїв користувачів |

Продовження табл. 1.1

| 1 | 2 | 3 | 4 | 5 | 6 |
|---|-----------------|--|---|--|--|
| 3 | Навчання кадрів | Cornerstone + EdCast, Degreed, Filtered Content Intelligence, WalkMe ActionBot | Аналіз бібліотек контенту для навчання персоналу, пропозиції і рекомендації | Мінімізація витрат на навчання, індивідуалізація програм підготовки кадрів | Важкість налаштувань, обмеженість у виборі програм |

Більшість наведених інструментальних засобів є органічним синтезом послідовних трансформацій практичних інструментів для управління персоналом організації, що історично передують інтегрованої організаційній культурі інших організаційних культур, наприклад, як у [6], або більш глибоке, з врахуванням специфіки організації, як у [8]. І з цього погляду відсутність у тій чи іншій національній чи корпоративній організаційній культурі історично попередніх і модернізованих елементів усіх трьох організаційних культур – бюрократичної, органічної та підприємницької – робить вельми проблематичною повноцінну реалізацію будь-якого з наведених інструментів через недостатньо ефективну організацію діяльності, неадекватного морально-психологічного настрою або (і) низького рівня професіоналізму. Все зазначене пояснює стовпець недоліків у табл. 1.1.

Взагалі можливість використання сучасних інструментальних засобів, що дозволяють автоматизувати роботу з персоналом, в тому числі ті, що використовують елементи ШІ, стала можливою завдяки зміни відношення до управління людьми в організації: від управління персоналом до управління кадровим ресурсом, що у підсумку призвело до підходу сприймати персонал організації, як кадровий потенціал або капітал [9]. Капітал потребував управління і вимагав зміни моделі кадрового менеджменту:

1) Перша модель. Менеджер з персоналу як піклувальник своїх працівників, що піклується про здорові умови праці та сприятливу морально-

психологічну атмосферу на підприємстві. Ця патерналістська модель перегукується з соціал-реформістським ідеям кінця XIX – початку XX ст. та знайшла своє органічне втілення у численних варіаціях доктрини людських відносин. Посадовий статус менеджера з персоналу цієї моделі досить низький: це клерк, який має підготовку в галузі промислової соціології (або психології) і допомагає лінійним керівникам проводити ефективну політику корпорації щодо найманих працівників. Рівень автоматизації процесів управління – низький. В основному використовуються комп'ютерні програми для організації діловодства;

2) Друга модель. Менеджер з персоналу як спеціаліст із трудових договорів (контрактів), включаючи колективні договори. У великих організаціях, де використовують масову низькокваліфіковану працю, менеджер з персоналу відіграє подвійну роль: здійснює адміністративний контроль над дотриманням найманими працівниками умов трудового договору та веде облік посадових переміщень; регулює трудові відносини у процесі переговорів із профспілками. Виконання цих функцій вимагає, як правило, юридичної підготовки. Рівень автоматизації процесів – середній. В роботі використовуються різноманітні інформаційно-довідкові системи, комунікаційні технології, бази даних;

3) Третя модель. Менеджер з персоналу як архітектор кадрового потенціалу організації, що грає провідну роль у розробці та реалізації довгострокової стратегії корпорації. Його місія – забезпечити організаційну та професійну відповідність складових кадрового потенціалу корпорації. Він входить до складу її вищого керівництва і має підготовку у сфері управлінського знання – управління людськими ресурсами. Рівень автоматизації процесів – високий. В роботі використовуються інформаційні технології, які дозволяють збирати, систематизувати, обробляти інформацію, дистанційно керувати складними процесами контролю якості кадрового потенціалу, приймати обґрунтовані рішення щодо добору кадрів, підвищення кваліфікації, сегментації персоналу.

Саме на автоматизацію третьої моделі кадрового менеджменту були зорієнтовані роботи дослідників ШІ, які працювали над питаннями впливу штучного інтелекту при його використанні в управлінні складними організаційними системами та, безпосередньо, зміни розуміння менеджменту при використанні сучасних інформаційних технологій з використанням елементів ШІ. Багато робіт присвячено алгоритмам штучного інтелекту в управлінні персоналом, новим підходам, методам та моделям. Серед дослідників названих напрямків можна вказати: R. Duchessi [10], D. O'Keefe. [11], V. Kolbjornsrud [12], M. El Khatib [13], O'Leary [14], H. McGrath [15], J. Ndiaye [16], B. M. Mohsen [17], O. Брінцеву [18], Б. Логвіненко [7], А. Матвійчук [19], О. Піжук [20].

Окремою низкою йдуть роботи дослідників, в яких представлені глибинні процеси застосування моделей та інструментальних засобів для прийняття рішень з відбору кадрів в межах новітніх інформаційних технологій, в різних сферах господарювання, зокрема, і стосовно організації управління людським капіталом підприємства, як частиною загального капіталу бізнесу. Серед таких робіт серед закордонних дослідників можна вказати монографії G.Luger [21], D. Poole [22], S. J. Russell [23].

Серед дослідників застосування штучного інтелекту в управлінні персоналом організацій в Україні окремим рядком йдуть роботи А.І. Шевченка. Зокрема, робота, що охоплює застосування ШІ в комунікаційній сфері [24], змістовне дослідження разом з Полехіним В.Д., що стосується саме розумової діяльності людини і відтворенні цих механізмів через ШІ [25], а також роботи, присвячені розробці графічного інтерфейсу програм, що використовують елементи ШІ [26] та підходи до організації дистанційного навчання за допомогою ШІ [27].

Аналізуючи наведені роботи, можна чітко означити потреби, які варто вирішувати за допомогою нових моделей та інструментальних засобів автоматизації процесів відбору кадрів з використанням Інтернет-платформ на сучасних підприємствах:

1) На відміну від управління персоналом, управління людськими ресурсами переорієнтовано з потреб персоналу на потреби самої організації в робочій силі, і пріоритети кадрового менеджменту визначаються насамперед результатами функціонального аналізу наявних і проєктованих робочих місць, а не існуючим кадровим потенціалом організації. Тобто, перша потреба, яка може вирішуватися через додатки на Інтернет-платформах – функціональний аналіз потреб організації та запуск технології відбору кадрів за виявлених потреб;

2) Поява стратегічного виміру в управління людськими ресурсами робить кадрову політику активнішою на відміну пасивної і реактивної політики, характерної традиційних моделей управління персоналом. Друга потреба – використання новітніх інструментальних засобів, наприклад, з використанням елементів ШІ, для постійного розвитку персоналу без задіяння додаткових кадрів для навчання та підвищення кваліфікації, навчання без відриву від місця роботи шляхом проходження тестування та перевірки знань через інтернет-платформи;

3) Відбувається переорієнтація системи кадрового менеджменту на індивідуальну роботу з персоналом, а, отже, перехід в управлінні персоналом з колективістських цінностей на індивідуалістичні. Тут сучасні моделі для прийняття рішень за умов невизначеності допоможуть вирішити проблему індивідуального кар'єрного зростання, постійного індивідуального моніторингу знань та вмінь і побудові траєкторії власної освіти на шляху кар'єрного розвитку.

Системи управління кадровим потенціалом здебільшого передбачають застосування на підприємствах сильної та адаптивної корпоративної культури, що стимулює атмосферу взаємної відповідальності найманого працівника та роботодавця, прагнення всіх працівників корпорації зробити її «найкращою компанією» за рахунок підтримки ініціативи на всіх рівнях організації, постійних технічних та організаційних нововведень, налагоджених прямих та

зворотних комунікацій [28]. Така технологія (див. табл. 1.1) дає синергетичний ефект, якщо в організації дотримуються наступних умов:

- відносно добре розвинена система адаптації до зовнішнього та внутрішнього ринку праці (індивідуальне планування кар'єри, підготовка та перепідготовка персоналу, стимулювання професійного зростання та ротації кадрів): Resume Matcher (SAP), TalentTech Sever.AI, Degreed, Filtered Content Intelligence;

- є гнучкі системи організації робіт (гуртки якості, автономні робочі групи): Hurma System, Cornerstone + EdCas, WalkMe ActionBot;

- використовуються системи оплати, побудовані на принципах всебічного обліку персонального вкладу (у тому числі й самим працівником) та (або) рівня професійної компетентності (знання, уміння та навички, якими реально опанували працівники): Hurma System, Workday;

- підтримується досить високий рівень участі окремих працівників та робочих груп у розробці та прийнятті управлінських рішень щодо їх повсякденної роботи: Resume Matcher (SAP), Veriato 360, Cornerstone + EdCas, WalkMe ActionBot;

- застосовується практика делегування повноважень підлеглим: Resume Matcher (SAP), Workday, Yva.ai 3.0, WalkMe ActionBot;

- функціонує розгалужена система організаційної комунікації, що забезпечує дво- та багатосторонні вертикальні, горизонтальні та діагональні зв'язки всередині організації: TalentTech Sever.AI, Veriato 360, Isaak Status Today by Glickon.

Огляд джерел за темою використання штучного інтелекту в системах відбору, сегментації та навчання кадрів показав, що існує ряд невирішених питань, зокрема:

- наявні інструменти управління відбором та розвитком персоналу пристосовані до використання у великих корпораціях, враховують особливості таких підприємств, при застосуванні на інших підприємствах або

використанні у рекрутингових центрах, деякі функції стають обмеженими або не придатними для налаштування;

– недостатня цифровізація населення не дозволяє використовувати ряд інструментів з управління кар'єрою, навчанням, професійним розвитком, на власних інформаційно-комунікаційних пристроях;

– більшість наведених інструментів управління добром та розвитком персоналу розроблені під конкретні процеси управління, не мають універсального характеру застосування;

– наявні приклади використання елементів ШІ в практиці управління персоналом мають описовий, загальний характер, що обумовлено комерційною таємницею, а на українських підприємствах взагалі таких прикладів не оприлюднено.

Тобто, сучасна практика використання новітніх моделей та інструментальних засобів для прийняття рішень за умов невизначеності при автоматизованому відборі персоналу за допомогою Інтернет-технологій потребує універсального інструменту, нескладного в керуванні, із забезпеченням кросплатформеності, із зрозумілим інтерфейсом та гнучкою технологією адаптації під потреби підприємства будь-якого рівня – від маленької компанії до корпорації. Подібне можна реалізувати шляхом переносу та адаптації деяких елементів базової платформи вікон, значків, меню та вказівок (WIMP) та підходів, використовуваних в теорії автоматів, у веб-середовище.

1.2 Керуючі впливи, обмеження та вхідні параметри при розробці моделей та інструментальних засобів для прийняття рішень з відбору кадрів за умов невизначеності

Розробляючи моделі та інструментальні засоби для прийняття рішень з відбору персоналу, в якій використовуватимуться сучасні підходи, наприклад, з використанням елементів штучного інтелекту, відбувається вплив ззовні на деякий об'єкт керування. У даному випадку об'єктом керування виступає саме технічний пристрій, який реалізує алгоритм чи то відбору необхідного фахівця, чи то визначення його рівня підготовки з наступним запуском задачі з навчання. Ці впливи надають поштовх системі, як позитивний, так і негативний, бо можуть мати деструктивний характер, в результаті чого алгоритми функціонування можуть почати працювати не вірно.

Параметри на вході об'єкта керування характеризують його початковий стан і у процесі роботи повинні цілеспрямовано змінюватися або підтримуватися постійними. Це є керовані величини, які залежать від вхідних впливів. Наприклад, на вході об'єкта управління можуть бути відео, текст та дія користувача, тож вхідними впливами можуть виступати швидкість потоку відео, параметри обміну даними в мережі, затримка сигналу при вводі тексту з клавіатури. Саме тут вхідні впливи розподіляються на вхідні і збуджуючі: одні відповідають алгоритму керування, інші заважають.

Керуючим впливом в системі, що аналізується, виступає швидкість обміну даними в мережі. Тоді обмеженням системи підбору та навчання кадрів виступатиме сукупність параметрів – технічних характеристик комунікаційної техніки, сервера та мережі. Крім того, суттєвими обмеженнями можуть стати ергономічні фактори, які або дозволяють користувачу швидко комунікувати зі ШІ, або обмежать цю можливість.

Виходячи із наведених вище невирішених питань, врахувати подібні керуючі впливи, обмеження та вхідні параметри можна при реалізації такої

системи у веб-середовищі – на Інтернет-платформі. Одним із варіантів реалізації може виступати 3D резюме.

3D резюме є новим поняттям в процедурах підбору, сегментації та навчанні кадрів. Хоча деякі організації вже використовують цей інструмент, проте чіткого визначення 3D резюме поки не існує. Проте, виходячи з ознак тримірного простору, як представлення деякого об'єкта в тривимірній Декартовій системі координат [29, 30] та поняття резюме, як документа про професійні досягнення особи [31], можна визначити 3D резюме як електронний документ (наприклад, HTML документ), який представляє професійні досягнення особи в трьох вимірах (інтерактивне представлення – текстові документи, співбесіда з відеозаписом та тестування в режимі онлайн).

Для взаємодії людини і машини під час проходження 3D резюме повинен бути розроблений спеціальний інтерфейс, який не лише дозволяє виконати завдання, а й стає певним завданням «нульового» рівня, яке визначає здатність людини зорієнтуватися та спланувати етапи проходження резюме.

За основу розробки такого інтерфейсу користувача можна використати WIMP (Windows, Icons, Menus, Pointing Device), як засіб взаємодії через вікна, меню та графічні елементи інтерфейсу 3D резюме. WIMP наразі є дещо застарілою технологією [32], проте деякі елементи випробувані часом та надійно зарекомендували себе у процедурах навчання для зменшення когнітивного навантаження на користувача в період активного отримання знань [33].

Базовий інтерфейс WIMP – інтерфейс користувача «людина-комп'ютер». Такий інтерфейс традиційно [34] має декілька типів, з яких для сучасної розробки [35] можна використати окремі елементи:

– текстовий блок (TUI – Text User Interface), зокрема, різновид у вигляді консолі (інтерфейсу командного рядка), який взаємодіє з користувачем за допомогою монітору і клавіатури, у тому числі – віртуальної;

– графічні елементи (GUI – Graphical User Interface), якими можуть виступати різноманітні інтерактивні кнопки, що слугують для переходу між завданнями на екрані монітору. Ці елементи тісно переплітаються елементами інтерфейсу (RUI), тобто, роботою через сенсорний екран планшету або смартфона;

– використання голосових команд або голосового вводу (VUI – Voice User Interface) для введення відповідей на окремі завдання або для введення тексту, ідентифікації користувача за голосом;

– природний інтерфейс користувача (NUI – Natural User Interface), який передбачає роботу із сенсорним екраном, створення середовищ з ефектом присутності, обробку жестів, міміки, різноманітних відгуків користувача на представлену задачу чи ситуацію. Саме у цьому випадку можливе використання ШІ для обробки зазначеної інформації на вході.

В останньому випадку можна говорити про спеціальний прикладний інтерфейс, пристосований до завдань конкретної технічної системи. І у цьому випадку обов’язковим є дослідження ефективності інтерфейсу через показник використовності або usability [36]. Основними показниками тут виступатимуть швидкість роботи користувачів, кількість помилок при цьому, швидкість навчання та задоволеність від роботи.

Подібне можна формалізувати, виходячи з [37], при цьому враховуючи, що робота користувача у веб-середовищі має свою специфіку [38]. При заповненні 3D резюме користувач несе деяке інтелектуальне навантаження $(IH)(R_x)$ при реалізації задачі R_x в системі 3D резюме:

$$IH(R_x) = \sum_{s(x)} \sum_{g(x)} m_{s(x),g(x)} r_{s(x),g(x)}, \quad (1.1)$$

де:

$m_{s(x),g(x)}$ – кількість виконаних операцій $g(x)$, в процедурі $s(x)$ задачі R_x ;

$r_{s(x),g(x)}$ – витрати праці операцій $g(x)$ процедури $s(x)$ задачі R_x в процесі її вирішення людиною (одиниці виміру обираються при організації досліджень).

Визначення того, яким чином система, що розробляється, може знизити інтелектуальне навантаження від роботи, дозволяючи повністю зацентувати увагу на вирішенні задачі можна через показник CAIQ, який наближує розуміння роботи системи саме до механізмів роботи ШІ [39]:

$$CAIQ = \frac{IH(R_0) - IH(R_x)}{IH(R_0)}, \quad (1.2)$$

де:

R_0 – деяка базова реалізація або аналогічна система, з якою відбувається порівняння.

Як можна зазначити з (1.2), результати можуть бути наступними:

- а) $IH(R_x) = 0$, $CAIQ=1$;
- б) $IH(R_0) = IH(R_x)$, $CAIQ = 0$;
- в) $IH(R_x) > IH(R_0)$, $CAIQ < 0$.

При отриманні останнього результату слід звернути увагу, що реалізація R_x є гіршою за базову.

Враховуючи типову схему проведення пошуку у веб-середовищі, включаючи логіко-семантичний апарат цього процесу [40], а також роботи до організації інтерфейсу користувача [41 – 43], можна навести узагальнену схему реалізації системи підбору персоналу, реалізовану у веб-середовищі (рис. 1.1).

На наведеній схемі під ідентифікацією розуміється контроль відповідності користувача введеним даним та візуальному зображенню. При ідентифікації підтягуються дані щодо поведінки користувача (відповіді, попереднє проходження навчання), які були отримані раніше. Будується схема

відповідності, яка дозволяє системі співставити інформацію, отриману раніше, з інформацією, яка буде отримана під час використання системи.

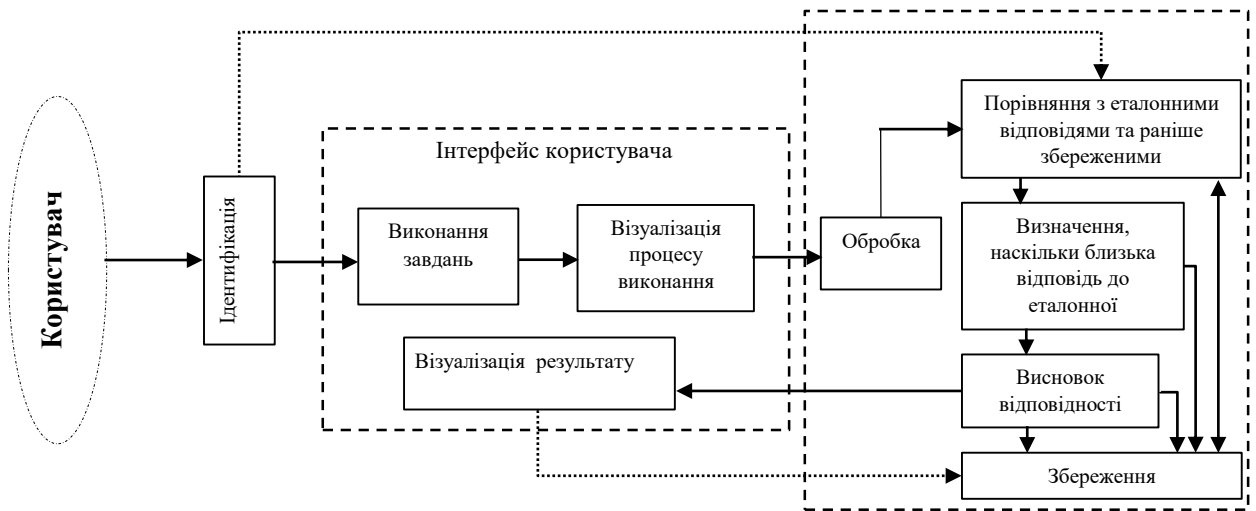


Рисунок 1.1 – Узагальнена схему реалізації системи підбору персоналу, реалізованої у веб-середовищі

В блоці обробки результатів за окремими вхідними параметрами ідентифікація відбувається декілька разів. Таким чином визначається прогрес/регрес користувача. Ідентифікація можлива з поповнюваною підказкою, і без підказки. Перший варіант полягає в тому, що в процесі надавання відповідей за завданням, можна використати вибір відповіді з переліку або вільну відповідь, тобто, в останньому випадку, ідентифікація відповідності відповіді може буди на основі еталонного слова чи сполучень окремих символів.

Якщо кожному слову-еталону відповідає певний елемент (або елементи) і для наведеного зразка існує відповідне слово-еталон, процес візуальної ідентифікації за цим питанням завершується. У протилежному випадку відповідь не приймається і тестування/відбір вважається не пройденим, або просто зараховується від'ємний результат і з'являється нове завдання для визначення рівня компетентностей.

Варіант реалізації подібної системи з підказкою є типовим, характерним для більшості інформаційних систем (ІС), починаючи від пошукових систем Інтернету та закінчуючи звичайними тестами з переліком можливих відповідей.

Безпосереднє визначення того, наскільки відповідь є близькою до еталонної можна за допомогою реалізації чотирьох алгоритмів корекції з ймовірностями наступних подій на основі обробки введених термінів або символів:

- помилка знайдена, ідентифікована, визначений відсоток вірної відповіді на заданому відрізьку балів;
- помилка виявлена, але ідентифікувати її не вдається, тому відкривається уточнююче питання;
- помилка виявлена, але вона ідентифікована не вірно, тому відкривається блок аналогічних питань для уточнення знань за темою;
- помилка не виявлена, відповідь є точною і їй присвоюється встановлений бал.

Подібні алгоритми можуть бути виконані послідовно [44]. Наприклад, приймаючи, що деяке еталонне слово (вірна відповідь) описується алфавітом $A_j = (a_1, \dots, a_i, \dots, a_m)$, де $(j = 1, \dots, N)$, помилка описується $B = (b_1, \dots, b_k, \dots, b_n)$, і до цього помилкового слова в системі відбувається пошук найближчої відповідності. Приймається умова, що це слово може відрізнитися від B на деяку множину помилок, що можуть коригуватися за допомогою наявного алфавіту.

На рис. 1.2 наведена загальна схема процесу порівняння на основі ймовірнісних методів [45] при змінах (помилках у відповіді) A_j . За допомогою $P(A)$ представлено оператор помилки – зміни або заміни наближеним терміном відповіді A_j . Помилка буде визначена за допомогою порівняння $P_A(A_j) = B$.

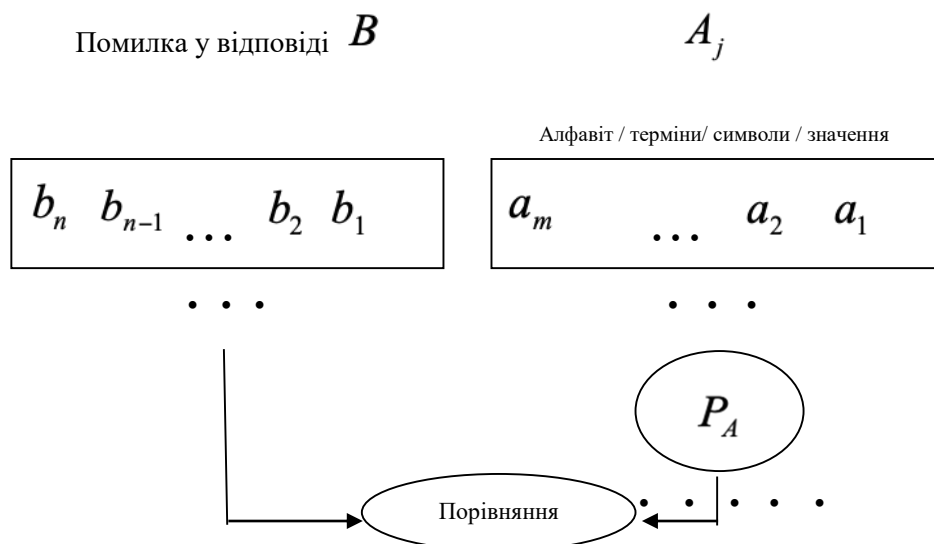


Рисунок 1.2 – Схема реалізації процесу порівняння отриманої відповіді із
еталонним значенням

Реалізація подібного аналітичного алгоритму пошуку відповідності може бути втілена за допомогою безпосереднього перебору з великою кількістю різнорідних величин на основі комбінаторного аналізу відповідно до заданих правил. За наведеною схемою імітаційну модель можна представити за допомогою мов C++ або Python.

Варто зазначити, що реалізуючи схему, наведену на рис. 1.1, при обробці результатів може виникнути необхідність створення структурованих запитів, в залежності від того, якого рівня база даних (БД) буде реалізована в системі роботи з персоналом: операційна, з використанням інструментарію витягу, перетворення та завантаження (ETL) з можливістю наступного завантаження оброблених даних до цільових аналітичних БД, Data Warehouse для аналітики, вітрини Data Marts для цільового аналізу, в тому числі тематичні. Розробка цільового інструментарію може стати високовитратною задачею, однак є багато стандартних програм [46], які можна використати для побудови запитів до БД. Наприклад, Query Builder від Oracle, Query Builder для СУБД SQL-сервер, Query Builder від Borland, Microsoft Query, Microsoft Access та інші. Вони доволі легко інтегруються та відповідають концепції людино-машинної

взаємодії [47] моделі людського інформаційного процесора спостереження за людиною і комп'ютером GOMS (Goals, Operators, Methods, Selection Rules) [48]. Загальна концепція GOMS передбачає опис цілей, операторів, методів, правил, що характеризують процедурні знання користувача, необхідні для виконання поставленої задачі, що повністю відповідає меті даної роботи.

1.3 Використання інструментів штучного інтелекту в сучасних системах прийняття рішень з відбору кадрів

Останнім часом [49] в автоматизованих системах відбору кадрів, які вийшли на абсолютно новий рівень, почали використовувати засоби штучного інтелекту. З 2015 року цей процес прискорився, навіть, з'явився вираз «tech meets HR» [50], що позначає переплетіння у єдине ціле технології та управління персоналом. Все це пов'язано з накопиченням та використанням великих даних (Big Data) та необхідністю швидко обробити отриману інформацію та відібрати кандидата, який найбільше відповідає організаційному замовленню.

Автоматизований підбір, сегментація та навчання персоналу – процес з багатьма керуючими впливами та великою кількістю вхідних параметрів, тому для реалізації моделей роботи з кадрами використовується хмарне середовище [51].

Як відомо [52], при відборі персоналу орієнтуються на розвиток організації на перспективу. Якщо є кадровий резерв – його досліджують, щоб визначити тих особистостей, кого можна швидко і без надлишкових витрат підготувати до роботи за наявною вакансією. Якщо відбувається зміна діяльності організації, то тоді задача кадрової служби – вибрати з наявного потоку респондентів фахівців, які мають всі необхідні навички та здатні швидко адаптуватися до змін. Звичайно ж, кожна організація зацікавлена

вияснити умови, за яких може відбутися відтік кадрів. Сучасні вимоги організацій та стрімкий розвиток науково-технічного прогресу ставлять нові виклики до автоматизації процесів відбору кадрів [53]. При цьому подібні системи повинні «навчитися» аналізувати особистий вибір кандидата, його власні цілі, плани, особливості, а також враховувати поточну ситуацію на ринку праці. Неодмінне питання, на яке повинна давати відповідь подібна система – оцінка і врахування перспектив розвитку кандидата на посаду та можливості його впливу на реалізацію стратегії організації.

Створення нових автоматизованих систем підбору кадрів обумовлена наступними факторами [52]:

- впровадження нової техніки та технології, зміни в комунікаційних можливостях, що особливо актуально проявилися під час карантинів, пов'язаних з коронавірусною хворобою, що призводить до корінних змін в організації і проведенні деяких видів робіт. За таких умов потрібна кваліфікація персоналу вже не може гарантуватися лише дипломом, а проведення співбесід та зустрічей обмежена вимогами ситуації, що склалася;

- ринок праці став не лише ринком високої конкуренції однієї країни. Тепер ринок праці перестав мати кордони. Країни, що мають високий рівень техніки і технології залучають кваліфікований персонал, що знаходиться далеко від місця роботи. Організації повинні швидко знаходити необхідний персонал на ринку праці, а також проводити необхідний контроль знань і вмінь наявного персоналу без необхідності викликати респондентів на бесіди в офіс;

- зміни у всіх сферах життя – головний елемент сучасності. Безперервні та швидкі зміни в технології та інформатиці вимагають безперервного самовдосконалення персоналу;

- для підприємства є більш ефективним і економічно вигідним вміння співробітників швидко пристосовуватися до змін в навколишньому світі, навички саморозвитку і вміння навчатися, ніж залучення весь час нових працівників у зв'язку із змінами у техніці і технології.

Врахувати все перелічене дозволяє 3D резюме. Проте ті компанії, що використовують цей інструмент, відносять цю автоматизовану систему до комерційної таємниці. Наприклад, 3D резюме використовується при відборі викладачів до Комп'ютерної Академії «Шаг» [54], проте доступ до системи 3D резюме стає можливим на дві доби тільки після персональної бесіди з відповідальним за відбір персоналу. Інструмент, використовуваний зазначеною компанією, передбачає приєднання текстових документів, відеоспівбесіду та вирішення задач програмування в режимі он-лайн. Кожне нове завдання резюме відкривається після виконання попередніх за ієрархією задач (наприклад, блиц-опитування перед вирішенням задач), по завершенню складання резюме відбувається оцінка знань та вмінь, а як тільки час складання резюме закінчується, доступ до системи обмежується.

Проте, задачу побудови 3D резюме для автоматизації підбору кадрів можна вирішити за допомогою теорії автоматів, яка застосовується в розробці штучного інтелекту [55] і може бути використана при розробці автоматизованого резюме при виконанні переходів від одного елементу вирішення задачі до іншого з врахуванням парадигми дискретного часу. Теорія автоматів використовується для вирішення задач машинного навчання, обробки природної мови та розпізнавання символів [56]. І рішенням такої задачі може стати мінімізація числа внутрішніх станів автомату. Для вирішення задачі використовується розбиття всіх станів абстрактного автомату на класи еквівалентних станів. Далі кожен клас еквівалентності замінюється представником даного класу, тобто, використовується один стан. Коли проведені всі перетворення, то отримується мінімальний автомат, число станів якого відповідає класам еквівалентності. Сама процедура мінімізації може бути проведена за алгоритмом Ауфенкампа-Хона [57]. А розглянути задачу можна через автомати Мура [58] та Мілі [59]. У даній роботі наведений приклад вирішення задачі автоматизації 3D резюме із застосуванням автомату Мура.

При розробці подібних систем на основі теорії цифрових автоматів варто пам'ятати, що загальна теорія цифрових автоматів підрозділяється на абстрактну і структурну [60].

Абстрактна теорія автоматів, не розглядаючи його структури, вивчає лише поведінку автомата щодо зовнішнього середовища. Саме тому основи абстрактної теорії автоматів дозволяють зробити систему, що здатна навчатися під впливом зовнішніх факторів. Абстрактна теорія цифрових автоматів є продовженням і деталізацією теорії алгоритмів. Різниця між абстрактної та структурної теоріями полягає в тому, що абстрактна теорія вивчає лише переходи, які зазнає автомат під впливом вхідних сигналів, і вихідні сигнали, які він при цьому видає, не вдаючись у питання кодування вхідних і вихідних сигналів і питання побудови автомата з елементарних схем. Таким чином, у загальному випадку синтез цифрових автоматів стає складним багатоетапним процесом, який використовують при розробці штучного інтелекту [56].

У випадку завдання структурного синтезу автомата з пам'яттю зводиться до знаходження спільних прийомів побудови структурних схем складних автоматів на основі композиції деяких елементарних автоматів, тобто пошуку певних способів їх з'єднання між собою.

Структурна теорія автоматів цікавиться як структурою самого автомата, так і структурою вхідних впливів і реакцій автомата на них. У ній вивчаються способи кодування вхідних впливів і реакцій автомата на них. Ця теорія є продовженням і розвитком абстрактної теорії автоматів. Спираючись на апарат булевих функцій і на абстрактну теорію автоматів, структурна теорія дає ефективні рекомендації з розробки реальних пристроїв обчислювальної техніки [60], які можуть бути використані у даній роботі.

Припустимо, що створюється базове 3D резюме для використання в своїй роботі агентством з найму персоналу. В основу розробки такого резюме буде покладений аналіз діяльності організації або галузі в цілому, бо на початку проведення відбору необхідно визначити, чи відповідає кандидат

мінімальним вимогам, які висуваються до персоналу. Тобто, на вході для первинного відбору персоналу в системі повинні бути завдання або питання, які дадуть відповіді на:

- скільки часу необхідно працівникові для виконання основних виробничих операцій, щоб визначити, чи відповідає наявний фахівець заявленим вимогам за технічними характеристиками;

- які виробничі операції можна згрупувати в більш загальне поняття робочого місця, щоб зрозуміти, наскільки фахівець орієнтується в сфері діяльності, на посаду в якій претендує;

- як організувати робоче місце таким чином, щоб збільшити продуктивність праці – для підтвердження базових практичних навичок;

- який режим роботи оптимальний для даного робочого місця – для визначення загальної компетенції;

- якими характеристиками (особливостями) повинен володіти працівник для виконання даної виробничої операції – для визначення рівня професійності;

- як може використовуватися інформація на даному робочому місці – для створення картини щодо можливості саморозвитку кандидата на посаду, його інноваційної компетентності, здатності до самоаналізу та самовдосконалення.

Подібні питання можуть бути розроблені і для інших сфер, які характеризують діяльність організації, як то: специфікація виробничого процесу, оцінка результативності праці, здатність до самовдосконалення, планування кар'єри, оплати та безпеки праці.

На кожне із зазначених питань можуть бути різні неоднозначні відповіді, які не підпадають під операції логічного типу даних, але можуть бути представлені у вигляді шкали відповідності певному встановленому критерію. Тобто, всі можливі відповіді на всі питання можуть бути представлені у вигляді таблиць з яких у підсумку можна вибрати скінченну кількість необхідних відповідей для вибору працівника.

Скінченна кількість відповідей дозволяє говорити про відслідковування змін завдань у 3D резюме в залежності від поточного стану інформації, що отримана від респондента. Тобто, для вирішення поставленої задачі необхідно провести мінімізацію абстрактного автомата Мура, заданого таблицею переходів і виходів, які відображають переходи між завданнями, в залежності від отриманої відповіді.

Варто зазначити, що для вирішення подібної задачі не обов'язково застосовувати підключення спеціальних обчислювальних служб через хмару. Зазначене можна реалізувати шляхом розробки спеціального сервісу на веб-платформі, який дозволить отримати відповіді респондента, їх систематизувати і надати кінцевому користувачу (менеджеру з персоналу) інформацію з критеріями відповідності чи невідповідності поставленим задачам.

Для цього слід встановити деякі факти і правила, на яке опиратиметься система таким чином, щоб наявні знання можна було використати в поточній ситуації. Тобто, повинна бути база фактів і база правил, і за допомогою правил відбувається використання фактів для створення обґрунтованого висновку [55]. Найпростіші правила можна виразити умовним відношенням «ЯКЩО, ТО». Тобто, використовуються звичайні оператори умови, наприклад, у JavaScript [61]. Це відповідає правилам роботи людського мозку: коли людина приступає до вирішення задачі, вона має певну множину інформації, з якої відбирається підмножина, що відповідає наявним умовам. Правильна реакція і відповідна відповідь реалізується через спрощення, коли блокуються думки стосовно інформації з інших множин, що не стосуються вирішення задачі (рис.1.3).

Зазначений конструкт стосовно використання штучного інтелекту в управлінні персоналом був оприлюднений наприкінці ХХ сторіччя [63]. Зокрема, зазначалося, що без врахування подібного процесу в управлінні персоналом результат створення автоматизованої системи кадрового менеджменту буде знівельовано, адже хтось сидітиме без діла, а хтось

робитиме другорядну роботу. І саме для створення програмної системи з використанням штучного інтелекту, до неї слід ввести всі елементи, які складають процес прийняття рішення людиною – мету, факти, правила, механізм вводу інформації і спрощення.

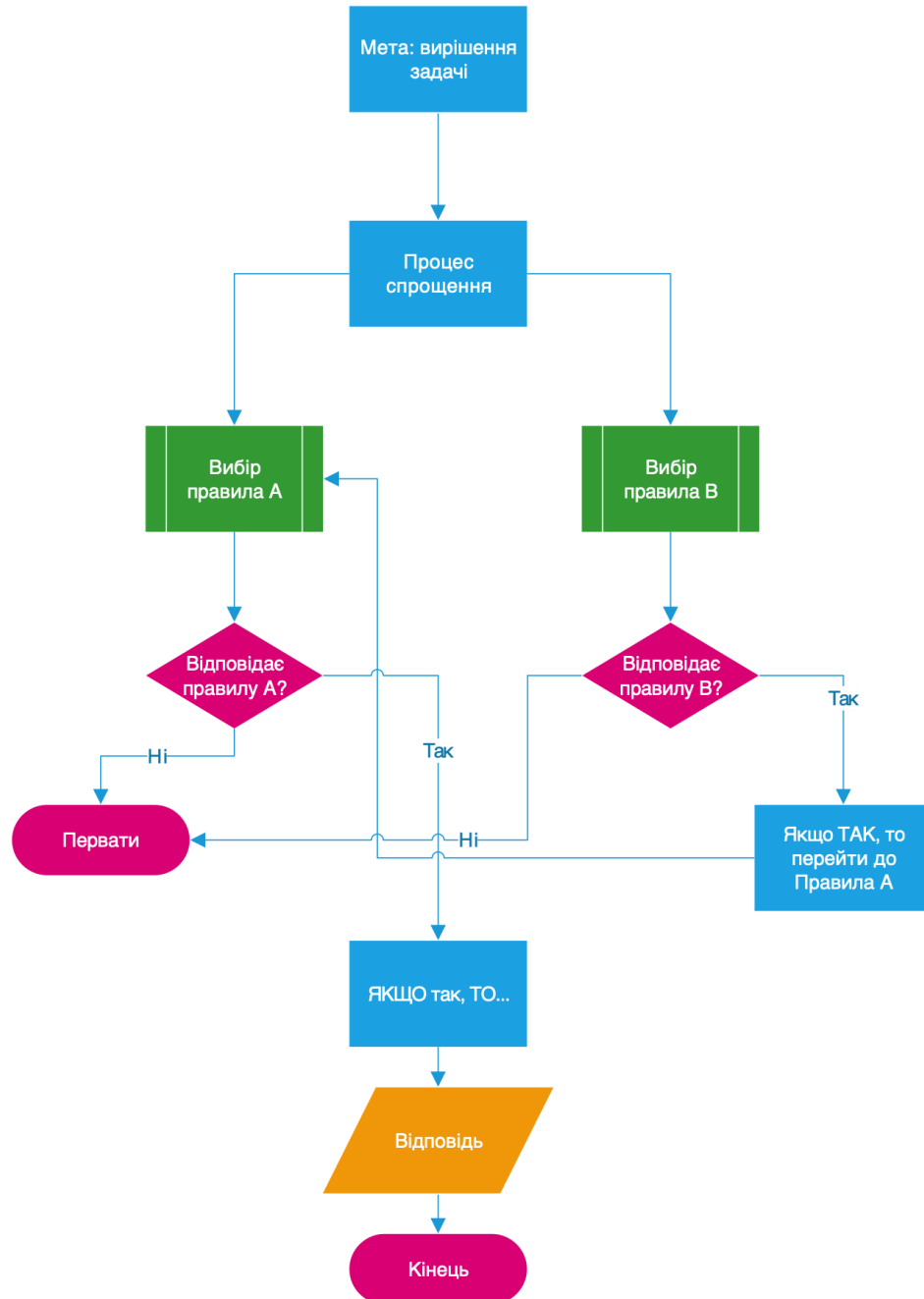


Рисунок 1.3. – Механізм спрощення при вирішенні задачі

Узагальнюючи викладене, варто зазначити, що створюючи систему підбору, сегментації та навчання кадрів з використанням елементів штучного

інтелекту, мета вже закладена у самій суті – за допомогою інформаційної технології знайти людину, що відповідає встановленим правилам зі спрощенням, яке стосується вибору підмножини інформації, що характеризує певну спеціальність та вимоги до спеціаліста на означену посаду. Залишається окреслити множину фактів, правила відбору на основі правил комбінаторики та реалізувати відповідний інтерфейс для втілення механізму вводу інформації, який дозволить отримати необхідну інформацію, до якої і будуть застосовані обрані правила для перевірки відповідності всіх фактів.

1.4 Технологія дослідження та методи вирішення поставлених задач

1.4.1 Технологія дослідження

Технологія дослідження роботи лежить в площині математичного моделювання та веб-програмування.

Початок будь-якого проекту з розробки моделі, яка буде закладена в основу програмного забезпечення, починається з двох питань:

1) Для кого? Тобто, для кого створюється ця система, яка цільова група користувачів та які їх основні вимоги.

2) Яким чином? Тобто, як будуватиметься процес розробки системи.

Наприклад, проект розробки інтегрованої інформаційно-аналітичної системи «Електронний Парламент» [63] тривав з 2011 по 2013 роки і складався з етапів розробки програмно-технічних засад, концепції, методології та архітектури, кожен з етапів тривав понад рік і теж складався з певного життєвого циклу: розпочинався обранням певної моделі розробки і завершувався здачею програмного модуля, технічної документації та представлення плану подальшого обслуговування та/або розвитку і удосконалення. В основу системи «Електронний Парламент» закладалися основні принципи, які ніяким чином не могли бути реалізовані одночасно: системності, модульності, головних задач, ієрархічності,

безперервного розвитку, розмежування доступу до інформації, моніторингу, санкціонованого доступу до інформації, єдиного інформаційного середовища, зворотного зв'язку, наочного представлення інформації і можливості вирішувати ситуаційні задачі. А все це повинне було бути об'єднане дружнім інтерфейсом користувача. З першого погляду здається, що деякі принципи неможливо поєднати в одній системі, бо виникає деяке протиріччя (наприклад, розмежування доступу до інформації і єдине інформаційне середовище). Але якщо до розробки застосувати модель, яка була обрана групою розробників, а саме, – модель водоспаду, то виявилось, що можна втілити всі зазначені принципи в одній системі.

Тому технологія дослідження роботи базується на принципах та математичних підходах, закладених науковою школою Трофимчука О.М. [64 – 65], на роботах і розробках співробітників Інституту телекомунікацій і глобального інформаційного простору НАН України – Довгого С.О. [63], Гуляєва К.Д. [66], Копійки О.В. [67].

Автоматизація роботи з кадрами невідривна від теорії і практики сучасного менеджменту. Тому базою для побудови моделей та алгоритмів виступають класичні роботи з менеджменту таких дослідників, як Drucker P.F. [6], Johnason P. [68], Klerck G. [69], Armstrong M. [70], Obedgiu V. [71] та інші.

Розробка системи підбору кадрів має за основу огляд та деталізацію підходів до веб-програмування, управління проектами, теорії алгоритмів таких дослідників, як Фрімен Е. [61], Лоусон Б., [72], Філліпс Дж. [73], Хопкрофт Дж. [74].

Обрана технологія дослідження та проведений аналіз літературних джерел дозволяє визначити методи реалізації дисертаційної роботи.

1.4.2 Методи реалізації роботи

Дисертаційна робота базується на застосуванні теорії автоматів для розробки системи відбору кадрів. Цифровий автомат можна трактувати як

пристрій, який здійснює прийом, збереження та перетворення дискретної інформації за деяким алгоритмом.

Абстрактний цифровий автомат є математичною моделлю дискретного керуючого пристрою.

Абстрактний цифровий автомат визначається множиною, що складається з шести елементів:

$$S = \{X, A, Y, \delta, \lambda, a_0\}, \quad (1.3)$$

де:

$X = \{x_1, x_2, \dots, x_n\}$ – множина вхідних сигналів (вхідний алфавіт);

$Y = \{y_1, y_2, \dots, y_m\}$ – множина вихідних сигналів (вихідний алфавіт);

$A = \{a_0, a_1, a_2, \dots, a_N\}$ – множина станів (алфавіт станів);

a_0 – початковий стан ($a_0 \in A$);

δ – функція переходів автомата, що задає відображення $(X \times A) \rightarrow A$, тобто ставить у відповідність будь-якій парі елементів декартового добутку $(X \times A)$ елемент множини A ;

λ – функція виходів автомата, що задає або відображення $(X \times A) \rightarrow Y$, або відображення $A \rightarrow Y$ [75].

Іншими словами, функція переходів δ у (1.3) показує, що автомат S , перебуваючи в деякому стані $a_j \in A$, при появі вхідного сигналу $x_j \in X$ переходить у якийсь стан $a_p \in A$. Це можна записати:

$$a_p = \delta(a_j, X_j). \quad (1.4)$$

А враховуючи (1.4), можна зазначити, що функція виходів показує, що автомат S , перебуваючи в деякому стані $a_j \in A$, при появі вхідного сигналу $x_j \in X$ видає вихідний сигнал $y_k \in Y$. Це можна записати:

$$y_k = \lambda(a_i, X_j). \quad (1.5)$$

Поняття стан у визначення автомата було введено в зв'язку з тим, що часто виникає необхідність в описі поведінки систем, виходи яких залежать не тільки від стану входів в даний момент часу, але і від деякої передісторії, тобто від сигналів, які надходили на входи системи раніше. Стан як раз і відповідає деякій пам'яті про минуле, дозволяючи усунути час як явну змінну і виразити вихідні сигнали як функцію станів і входів в даний момент часу.

Викладене можна використати для реалізації задач дослідження наступним чином:

- скласти таблицю переходів і виходів абстрактного автомата;
- провести мінімізацію числа внутрішніх станів автомата із застосуванням алгоритму Ауфенкампа-Хону, побудувати граф переходів автомата;
- побудувати скорочені і повні таблиці переходів тригерів і відповідні їм матриці переходів тригерів, закодувати вхідні і алфавіти автомату;
- записати канонічні вирази для функцій збудження елементів пам'яті і функцій виходів автомата та побудувати функціональну схему автомата, який буде лежати в основі системи підбору, сегментації та навчання кадрів. Сам алгоритм інформаційної технології базуватиметься на основі комбінаторного аналізу [76], здійснюючи перестановку, подібну до того, як це здійснюється за допомогою абстрактного автомату.

Коли респондент відповідатиме на питання, що відкрилося в системі, система не лише запам'ятає вибір відповіді та співвідноситиме її з зареєстрованим респондентом, а й відкриє наступне питання в залежності від того, наскільки правильною була отримана відповідь. При покращенні результату опитування система відкриватиме більш складні завдання, при погіршенні – менш складні, для того, щоб вірно визначити рівень підготовки претендента на певну посаду.

Проте успішна робота респондента із системою підбору, сегментації та навчання кадрів залежить також від зручності та трудомісткості інтерфейсу. В роботі передбачається розробити відповідний інтерфейс системи роботи з кадрами, оцінка якого буде вивірена за допомогою Keystroke-Level Model (KLM) [48], або моделі натискання клавіш [42]. За допомогою моделі GOMS-KLM можна розрахувати необхідний для користувача час, за який він може виконати поставлені завдання. Існують вивірені дії натискання клавіш:

K – натискання однієї клавіші на клавіатурі (0,12 – 1,2с; стандарт користувача 0,28с);

$T(n)$ – набирання деякої послідовності n символів на клавіатурі ($n \cdot K$);

P – вказівка мишею (наприклад, натискання кнопки). Стандарт для користувача – усереднене значення, яке дорівнює 1,1с;

B – натискання кнопки миші (0,1с);

BB – клік кнопкою миші (0,2с);

H – рух рукою від клавіатури до миші (0,4с);

M – роздум над запитанням (0,6 – 1,35с). Усереднений стандарт користувача складає 1,2с;

$W(t)$ – очікування відповіді системи. Сучасний користувач працює в термінах 2 – 3 с.

Застосування методу KLM для оцінки часу роботи респондента із системою, що розробляється, на основі інтерфейсу, з яким респондент працюватиме, базуватиметься на наступних кроках:

– визначення секторів інтерфейсу, де можна перерахувати дії користувача;

– прогнозування шляху користувача при відповіді на запитання;

– визначення дій та місць, де користувач повинен подумати, перед тим, як надати відповідь;

– сума часу окремих дій і є оцінка часу роботи користувача із системою.

Моделювання, програмування та проведення апробації системи є методами практичних розділів даної роботи. Алгоритмізація окремих процесів

дозволить пояснити особливості застосування елементів штучного інтелекту при відборі спеціаліста за заданими роботодавцем вимогами. Апробація полягає у проведенні натурального експерименту, який дозволить оцінити ефективність роботи користувача із системою, а також можливості системи запам'ятовувати відповіді і правильно вибирати наступні завдання згідно попередньої оцінки системою знань респондента.

1.5 Висновки за розділом 1

Теоретичне дослідження основ розробки моделей та інструментальних засобів з відбору кадрів для прийняття рішень за умов невизначеності на Інтернет-платформах, дозволила сформулювати наступні узагальнення:

1) сучасні парадигми кадрового менеджменту розглядають трудові ресурси, як кадровий потенціал організації. Відповідно, це вимагає нових методів і підходів до його формування, а загальна цифровізація суспільства – застосування сучасних технологій в роботі. Саме тому зазначена тема роботи є актуальною;

2) при доволі широкому представленні робіт автоматизації процесів відбору кадрів, роботи щодо реалізації інструментальних засобів відбору на Інтернет-платформах залишилися на рівні теоретичних напрацювань. Саме тому для реалізації поставлених в роботі задач пропонується реалізація такого інструментального засобу, як 3D резюме. Поняття 3D резюме пропонується розглядати через документ HTML-формату, який представляє професійні досягнення особи в трьох вимірах через текст, співбесіду з відеозаписом та тестування в режимі он-лайн;

3) робота буде реалізована за допомогою застосування теорії цифрових автоматів, елементів теорії ймовірності та математичної логіки для вирішення задач вибору, у поєднанні з веб-програмуванням та розробкою відповідного

інтерфейсу користувача. Сутність розробки полягатиме у надання респондентом відповідей системі, яка не лише запам'ятає вибір відповіді та співвіднесе її із зареєстрованим респондентом, а й відкриє наступне питання в залежності від того, наскільки правильною була отримана відповідь;

4) обрана методика дослідження відповідає поставленим задачам дослідження і дозволяє досягти мети роботи, якою є вирішення наукової задачі з використання засобів штучного інтелекту при розробці сучасних систем підбору, сегментації та навчання кадрів. Наведене дозволило обрати загальнотеоретичні засади, підходи та методи для досягнення мети роботи.

5) Матеріали, що подані в розділі, опубліковані в [77 – 80].

РОЗДІЛ 2

РЕАЛІЗАЦІЯ ДІЙ ПОВТОРУ ТА ВИБОРУ ДЛЯ РЕАЛІЗАЦІЇ ІНСТРУМЕНТАЛЬНОГО ЗАСОБУ З ВІДТВОРЕННЯ ІНТЕЛЕКТУАЛЬНОЇ ДІЯЛЬНОСТІ ЛЮДИНИ

2.1 Використання абстрактного автомату для створення механізму відтворення людської поведінки

Враховуючи принципи організації та взаємодії відкритих систем [81] з чітким розділенням керування системою та керуванням вирішення прикладних задач, та використовуючи технології організації сучасних телекомунікацій [82], можна реалізувати тісну взаємодію інформаційно-комунікаційних технологій та штучного інтелекту [83] у веб-сервісі з чітко означеними задачами.

Орієнтуючись на моделі (1.3) – (1.5), можна зробити висновок про те, що абстрактний автомат функціонує в дискретному часі $t = 0, 1, 2, \dots$ і переходить між станами здійснюються миттєво. У кожен момент t дискретного часу автомат знаходиться в певному стані $a(t)$ з множини A станів автомату, причому в початковий момент часу $t = 0$ він завжди знаходиться в початковому стані a_0 . У момент часу t перебуваючи в стані $a(t)$, автомат здатний сприйняти на вхідному каналі сигнал $x(t) \in X$ і видати на вихідному каналі сигнал $y(t) = \lambda(a(t), x(t))$, переходячи в стан $a(t + 1) = \delta(a(t), x(t))$. Залежність вихідного сигналу від вхідного стану вказує про наявність пам'яті.

Останнє і використовується при розробці штучного інтелекту для опису поведінки нервової системи біологічних суб'єктів [55].

Подібне функціонування автомату дозволяє робити повтор дії та вибір наступного кроку (питання або завдання) в залежності від правильності попередньої дії (істини або хибності у вирішенні поставленого завдання).

За способом формування функції виходів виділяють три типи абстрактних автоматів: автомат Мілі, автомат Мура, С–автомат [60]. Використовуючи [58] можна вивести основні характеристики цих автоматів, щоб обрати ті автомати, що найбільше підходять для виконання задач роботи.

Автомат Мілі характеризується системою рівнянь:

$$\begin{aligned} y(t) &= \lambda(a(t), x(t)); \\ a(t + 1) &= \delta(a(t), x(t)). \end{aligned} \quad (2.1)$$

Автомат Мура можна описати, як:

$$\begin{aligned} y(t) &= \lambda(a(t)); \\ a(t + 1) &= \delta(a(t), x(t)). \end{aligned} \quad (2.2)$$

С–автомат:

$$\begin{aligned} y &= y_1 \cup y_2; \\ y_1(t) &= \lambda_1(a(t), x(t)); \\ y_2(t) &= \lambda_2(a(t)); \\ a(t + 1) &= \delta(a(t), x(t)). \end{aligned} \quad (2.3)$$

Якщо на вхід абстрактного автомата Мілі (2.1) або Мура (2.2), встановленого в початковий стан a_0 , подавати буква за буквою деяку послідовність букв вхідного алфавіту $x(0), x(1), \dots$ – вхідне слово, то на виході автомата будуть послідовно з'являтися літери вихідного алфавіту $y(0), y(1), \dots$ – вихідне слово. Для випадку С–автомата (2.3) на його виходах будуть з'являтися дві послідовності: $y_1(0), y_2(1), \dots$ та $y_2(0), y_2(1), \dots$. В абстрактному С–автоматі вихідний сигнал $y_2(t) = \lambda_2(a(t))$ видається весь

час, поки автомат знаходиться в стані $a(t)$. Вихідний сигнал $y_1(t) = \lambda_1(a(t), x(t))$ видається під час дії вхідного сигналу $x(t)$ при знаходженні S -автомата в стані $a(t)$.

Таким чином, найбільше для задачі, поставленій у роботі, підходять автомати Мілі та Мура, які найкращим чином дозволяють здійснити перетворення вхідних слів у вихідні слова.

Щоб задати кінцевий автомат S , необхідно описати всі елементи множини $S = \{X, A, Y, \delta, \lambda, a_0\}$. Існує кілька способів [84] завдання роботи автомату, але найбільш часто використовується табличний (матричний), графічний, аналітичний [57]. Для веб-реалізації найбільш зручним способом вважається табличний, бо дозволяє реалізацію за допомогою каскадних таблиць стилю CSS як у самому html-документі, так і у зовнішніх файлах.

При табличному способі автомат задається двома таблицями: таблицею переходів і таблицею виходів, або матрицею з'єднань. У клітинці таблиці переходів, що знаходиться на перетині рядка, зазначеного вхідним сигналом x_i , і стовпця зазначеного станом a_j , зазначається стан a_k , що є результатом переходу автомату зі стану a_j під впливом вхідного сигналу x_i за виразом $a_k = \delta(a_j, x_j)$.

Наприклад, заповнення таблиці переходів абстрактного повністю визначеного автомату, який покладено в основу вибору відповідей 3D резюме з вхідним набором ключових слів (термінів за спеціальністю) $X = \{x_1, x_2\}$ і алфавітом станів відповідей (слів, що максимально наближають відповідь до ключових слів) $A = \{a_1, a_2, a_2\}$.

Якщо абстрактний автомат частковий, то в клітинці таблиці його переходів, що знаходиться, на перетині рядка, зазначеної вхідним сигналом і стовпця зазначеного відповідним станом (за умови, що перехід у цей стан під дією даного вхідного сигналу не визначено) ставиться прочерк, і будь яке вхідне слово, що приводить до зазначеного переходу є забороненим [85]. Наприклад, якщо відповідь претендента при заповненні 3D резюме не

співпадає хоча б з одним з ключових слів, то наступне завдання 3D резюме не активується.

Заповнення інших клітин аналогічно випадку повністю визначеного автомата. Вид таблиці переходів не залежить від типу заданого автомата (автомат Мілі, Мура). Таблиці виходів автоматів Мілі та Мура мають відмінності, які варто враховувати при розробці систем підбору, сегментації та навчання кадрів.

Для цього варто зауважити, що таблиця виходів повністю визначеного автомата Мілі будується наступним чином: ідентифікація стовпців і рядків, а також формат таблиці відповідають таблиці переходів повністю визначеного автомата. У клітинці таблиці виходів, що знаходиться на перетині рядка, зазначеної вхідним сигналом x_j , і стовпця, зазначеного станом a_k , прописується вихідний сигнал u_m , який автомат видає, перебуваючи в стані a_k при наявності вхідного сигналу x_j , що визначається виразом: $u_m = \lambda(a_k, x_j)$.

Спрощений механізм практичного застосування автомату Мілі у веб-реалізації сервісу з визначення рівня підготовки персоналу та необхідності підвищення кваліфікації можна представити наступним чином: виникають завдання і варіанти відповідей, необхідно дати одну або декілька правильних відповідей, щоб у підсумку отримати необхідну кількість балів для продовження роботи, інакше буде прийняте рішення про необхідність підвищення кваліфікації. Для того, щоб дати правильну відповідь, необхідно здійснити перебір можливих правильних значень таким чином, щоб зберігався попередній стан або здійснювався перехід до питання, що веде на рівень вище. Але переходи між питаннями можуть вести як на рівень вище, так і на рівень нижче, що дозволяє точно описувати підготовку фахівця. При цьому кількість варіантів правильних відповідей – обмежена.

Використання автомату Мілі для зазначеної задачі може складатися з наступних кроків:

1) Крок 1 – визначення елементів вхідного алфавіту X , алфавіту станів A , вихідного алфавіту Y .

Вхідний алфавіт $X = \{x_1, x_2, x_3\}$ – складається з 3 літер (позицій, слів, словосполучень).

Алфавіт станів $A = \{a_1, a_2, a_3, a_4\}$ – складається з 4 літер.

Вихідний алфавіт $Y = \{y_1, y_2, y_3\}$ – складається з 3 літер;

2) Крок 2 – визначення функції переходів.

Функція переходів $\delta(a_1, x_1) = a_1$, свідчить про те, що автомат S , перебуваючи в стані a_1 при надходженні на його вхід вхідного сигналу x_1 переходить в стан a_1 , тобто стан не змінюється (зберігає попередній стан).

$\delta(a_2, x_1) = a_3$ – при надходженні вхідного сигналу x_1 здійснюється перехід $a_2 \rightarrow a_3$;

$\delta(a_4, x_3) = a_2$ – при надходженні вхідного сигналу x_3 здійснюється перехід $a_4 \rightarrow a_2$;

3) Крок 3 – визначення функції виходів.

Функція виходів $\lambda(a_1, x_2) = y_1$, свідчить про те, що при надходженні вхідного сигналу x_2 на вхід автомату, якщо він знаходився в стані a_1 , то на виході автомата формується вихідний сигнал y_1 ;

$\lambda(a_3, x_3) = y_3$; $\lambda(a_4, x_1) = y_3$;

4) Крок 4 – формування таблиці переходів заданого автомату Мілі. Сумісна таблиця переходів поєднуватиме таблицю переходів і таблицю виходів.

5) Крок 5 – складання матриці з'єднань заданого автомату. Матриця з'єднань представляє собою квадратну матрицю розміром $A \times A$ з об'єднанням таблиць переходів і виходів абстрактного автомату. Матриця з'єднань дозволяє співставити отримані відповіді респондента з базою результатів, необхідних для отримання потрібної кількості балів для підтвердження кваліфікації або висновку про необхідність підвищення кваліфікації.

Цей автомат можна задати аналітично четвіркою об'єктів $S = \{X, A, Y, F\}$, де F задає для кожного стану a_i автомата відображення $(X * A) \rightarrow (A * Y)$. Тоді алгоритм реалізації задачі складатиметься з наступних кроків:

- 1) Початок;
- 2) Вхідні дані: $X = \{x_1, x_2, x_3\}$; $A = \{a_1, a_2, a_3, a_4\}$; $Y = \{y_1, y_2, y_3\}$;
- 3) З'єднання: $F_{a1} = \{a_1(x_1/y_1), a_2(x_2/y_1), a_1(x_3/y_2)\}$;
- 4) З'єднання: $F_{a2} = \{a_3(x_1/y_3), a_4(x_2/y_3), a_2(x_3/y_3)\}$;
- 5) З'єднання: $F_{a3} = \{a_3(x_1/y_2), a_1(x_2/y_1), a_4(x_3/y_3)\}$;
- 6) З'єднання: $F_{a4} = \{a_3(x_1/y_2), a_1(x_2/y_1), a_2(x_3/y_3)\}$;
- 7) Закінчення алгоритму.

Тобто, в підсумку буде отриманий шуканий результат.

Кожному автомату Мілі існує еквівалентний йому автомат Мура [59]. Різниця між машинами Мілі і Мура полягає у тому, що у останнього вихідні символи визначаються тільки з його стану, на відміну від першого, де важливий і стан, і символи на вході. Саме тому, для виконання завдання роботи є цікавим автомат Мура.

2.2 Мінімізація числа внутрішніх станів автомату щодо задачі створення 3D резюме

Для виконання роботи за допомогою автомата Мура приймаємо, що він заданий таблицею переходів і виходів (табл. 2.1), що відображає переходи між завданнями, в залежності від отриманої відповіді.

Таблиця 2.1 – Таблиця переходів і виходів

| | | | | | | | | | | |
|----------------|----------------|----------------|----------------|----------------|-----------------|----------------|----------------|-----------------|-----------------|-----------------|
| У | у ₁ | у ₂ | у ₃ | у ₄ | у ₁ | у ₂ | у ₄ | у ₂ | у ₁ | у ₁ |
| A X | а ₁ | а ₂ | а ₃ | а ₄ | а ₅ | а ₆ | а ₇ | а ₈ | а ₉ | а ₁₀ |
| х ₁ | а ₁ | а ₃ | а ₆ | а ₄ | а ₇ | а ₅ | а ₄ | а ₉ | а ₇ | а ₁₀ |
| х ₂ | а ₅ | а ₁ | а ₃ | а ₆ | а ₁₀ | а ₉ | а ₈ | а ₁₀ | а ₉ | а ₅ |
| х ₃ | а ₇ | а ₄ | а ₅ | а ₂ | а ₈ | а ₈ | а ₁ | а ₃ | а ₁₀ | а ₇ |

Виконаємо розбивання $\pi_0 = \{B_1, B_2, B_3, B_4\}$. Для автомата Мура розбивання π_0 проводиться за вихідним сигналом:

$$B_1 = \{a_1, a_5, a_9, a_{10}\}, B_2 = \{a_2, a_6, a_8\}, B_3 = \{a_3\}, B_4 = \{a_4, a_7\}.$$

Будуємо таблицю розбиття π_0 (табл. 2.2).

Таблиця 2.2 – Таблиця розбиття π_0

| | B ₁ | | | | B ₂ | | | B ₃ | B ₄ | |
|----------------|----------------|----------------|----------------|-----------------|----------------|----------------|----------------|----------------|----------------|----------------|
| A | а ₁ | а ₅ | а ₉ | а ₁₀ | а ₂ | а ₆ | а ₈ | а ₃ | а ₄ | а ₇ |
| х ₁ | B ₁ | B ₄ | B ₄ | B ₁ | B ₃ | B ₁ | B ₁ | B ₂ | B ₄ | B ₄ |
| х ₂ | B ₁ | B ₁ | B ₁ | B ₁ | B ₁ | B ₁ | B ₁ | B ₃ | B ₂ | B ₂ |
| х ₃ | B ₄ | B ₂ | B ₁ | B ₄ | B ₄ | B ₂ | B ₃ | B ₁ | B ₂ | B ₁ |

За таблицею розбиття π_0 (табл. 2.2) виконується розбиття π_1 . При виконанні цього розбиття аналіз проводиться тільки всередині кожної окремої множини B_i :

$$\pi_1 = \{C_1, C_2, C_3, C_4, C_5, C_6, C_7, C_8, C_9\}$$

$$C_1 = \{a_1, a_{10}\}, C_2 = \{a_5\}, C_3 = \{a_9\}, C_4 = \{a_2\}, C_5 = \{a_6\}, C_6 = \{a_8\}, C_7 = \{a_3\}, \\ C_8 = \{a_4\}, C_9 = \{a_7\}.$$

З отриманого будується таблиця розбиття π_1 (табл. 2.3).

Таблиця 2.3 – Таблиця розбиття π_1

| | C ₁ | | C ₂ | C ₃ | C ₄ | C ₅ | C ₆ | C ₇ | C ₈ | C ₉ |
|----------------|----------------|-----------------|----------------|----------------|----------------|----------------|----------------|----------------|----------------|----------------|
| A | a ₁ | a ₁₀ | a ₅ | a ₉ | a ₂ | a ₆ | a ₈ | a ₃ | a ₄ | a ₇ |
| x ₁ | C ₁ | C ₁ | C ₉ | C ₉ | C ₇ | C ₂ | C ₃ | C ₅ | C ₈ | C ₈ |
| x ₂ | C ₂ | C ₂ | C ₁ | C ₃ | C ₁ | C ₃ | C ₁ | C ₇ | C ₅ | C ₆ |
| x ₃ | C ₉ | C ₉ | C ₆ | C ₁ | C ₈ | C ₆ | C ₇ | C ₂ | C ₄ | C ₁ |

За таблицею розбиття π_1 (табл. 3.) можна виконати розбиття π_2 .

$$\pi_2 = \{D_1, D_2, D_3, D_4, D_5, D_6, D_7, D_8, D_9\}.$$

$$D_1 = \{a_1, a_{10}\}, D_2 = \{a_5\}, D_3 = \{a_9\}, D_4 = \{a_2\}, D_5 = \{a_6\}, D_6 = \{a_8\}, D_7 = \{a_3\}, \\ D_8 = \{a_4\}, D_9 = \{a_7\}.$$

Розбиття π_2 повторює розбиття π_1 – процедура розрахунків може бути завершена. З кожного класу еквівалентності вибираємо по одному представнику цього класу. З множини $D_1 = \{a_1, a_{10}\}$ вибираємо a_1 , тому $a_1 \equiv a_{10}$. У таблиці переходів стовпець, який відповідає стану a_{10} викреслюємо, а в решті таблиці замінюємо a_{10} на a_1 . Отримуємо об'єднану таблицю переходів і виходів мінімального автомату (табл. 2.4).

Таблиця 2.4 – Об'єднана таблиця переходів і виходів мінімального автомату

| Y | y ₁ | y ₂ | y ₃ | y ₄ | y ₁ | y ₂ | y ₄ | y ₂ | y ₁ |
|----------------|----------------|----------------|----------------|----------------|----------------|----------------|----------------|----------------|----------------|
| A | a ₁ | a ₂ | a ₃ | a ₄ | a ₅ | a ₆ | a ₇ | a ₈ | a ₉ |
| X | a ₁ | a ₃ | a ₆ | a ₄ | a ₇ | a ₅ | a ₄ | a ₉ | a ₇ |
| x ₁ | a ₁ | a ₃ | a ₆ | a ₄ | a ₇ | a ₅ | a ₄ | a ₉ | a ₇ |
| x ₂ | a ₅ | a ₁ | a ₃ | a ₆ | a ₁ | a ₉ | a ₈ | a ₁ | a ₉ |
| x ₃ | a ₇ | a ₄ | a ₅ | a ₂ | a ₈ | a ₈ | a ₁ | a ₃ | a ₁ |

На підставі табл. 2.4 можна побудувати граф переходів автомата Мура (рис. 2.1), який фактично є графом 3D резюме.

Слід зазначити, що автомат Мура базується на входних діях. Для створення 3D резюме саме входні дії є вирішальними, бо у підсумку, за рядом визначених критеріїв, дають відповідь на запитання «ІСТИНА» або «ХИБА» для переходу до більш складних питань.

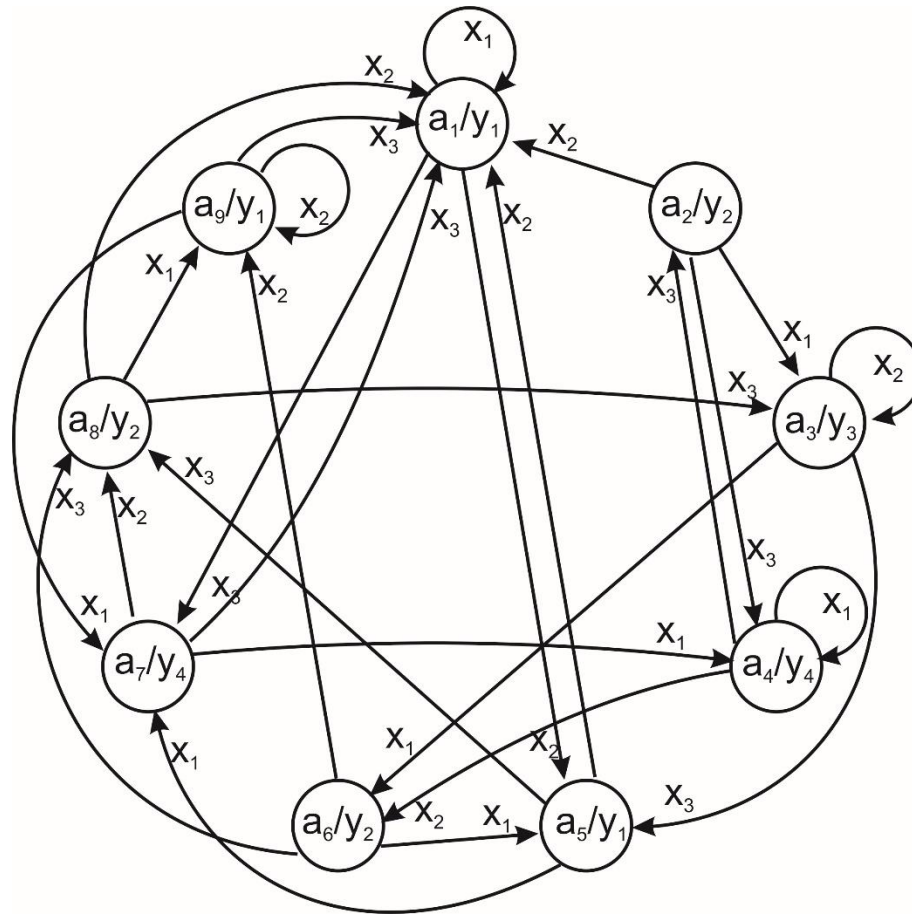


Рисунок 2.1. – Граф переходів автомата Мура, що візуалізує схему 3D резюме

Для того, щоб зрозуміти ці переходи та прибрати з 3D резюме відповіді на питання, які можуть запустити алгоритм відбору за повторним циклом, слід попередньо дослідити можливий алгоритм кодування станів та скласти функціональну схему автомату переходів.

Для тригера, заданого табл. 2.2 можна побудувати таблицю переходів (табл. 2.5) скорочену, з двома інформаційними входами (табл. 2.3, (а)) і повну таблицю переходів асинхронного тригера з двома інформаційними входами X і Y (табл. 2.5, (б)).

При побудові таблиць переходів синхронного тригера (додатково до входів X і Y вводиться вхід синхронізації C), слід мати на увазі, що при $C = 0$ внутрішній стан тригера не змінюється незалежно від станів входів X і Y , тобто $Q(t+1) = Q(t)$, а при $C = 1$ синхронний тригер функціонує як відповідний асинхронний.

Таблиця 2.5 – Таблиці переходів з двома інформаційними входами

| t | | (t+1) |
|---|---|-------------------|
| X | Y | Q |
| 0 | 0 | 0 |
| 0 | 1 | $Q(t)$ |
| 1 | 0 | $\overline{Q(t)}$ |
| 1 | 1 | 1 |

а) скорочена

| t | | | t+1 |
|---|---|---|-----|
| X | Y | Q | Q |
| 0 | 0 | 0 | 0 |
| 0 | 0 | 1 | 0 |
| 0 | 1 | 0 | 0 |
| 0 | 1 | 1 | 1 |
| 1 | 0 | 0 | 1 |
| 1 | 0 | 1 | 0 |
| 1 | 1 | 0 | 1 |
| 1 | 1 | 1 | 1 |

б) повна

З урахуванням зазначеного вище отримуємо скорочену (табл. 2.6, (а)) і повну (табл. 2.6, (б)) таблиці переходів синхронного тригера.

Тригер має чотири можливі варіанти переходів: "0-0", "0-1", "1-0", "1-1". З табл. 2.5 (б) бачимо, що цим переходам відповідають такі комбінації сигналів X і Y (табл. 2.7).

Таблиця 2.6 – Таблиця переходів синхронного тригера

| t | | | t+1 |
|---|---|---|-------------------|
| C | X | Y | Q |
| 0 | 0 | 0 | $Q(t)$ |
| 0 | 0 | 1 | $Q(t)$ |
| 0 | 1 | 0 | $Q(t)$ |
| 0 | 1 | 1 | $Q(t)$ |
| 1 | 0 | 0 | 0 |
| 1 | 0 | 1 | $Q(t)$ |
| 1 | 1 | 0 | $\overline{Q(t)}$ |
| 1 | 1 | 1 | 1 |

| t | | | | t+1 |
|---|---|---|---|-----|
| C | X | Y | Q | Q |
| 0 | 0 | 0 | 0 | 0 |
| 0 | 0 | 0 | 1 | 1 |
| 0 | 0 | 1 | 0 | 0 |
| 0 | 0 | 1 | 1 | 1 |
| 0 | 1 | 0 | 0 | 0 |
| 0 | 1 | 0 | 1 | 1 |
| 0 | 1 | 1 | 0 | 0 |
| 0 | 1 | 1 | 1 | 1 |
| 1 | 0 | 0 | 0 | 0 |
| 1 | 0 | 0 | 1 | 0 |
| 1 | 0 | 1 | 0 | 0 |
| 1 | 0 | 1 | 1 | 1 |
| 1 | 1 | 0 | 0 | 1 |
| 1 | 1 | 0 | 1 | 0 |
| 1 | 1 | 1 | 0 | 1 |
| 1 | 1 | 1 | 1 | 1 |

а) скорочена

б) повна

Таблиця 2.7 – Таблиця комбінування сигналів

| Q(t) | Q(t+1) | X | Y |
|------|--------|---|---|
| 0 | 0 | 0 | 0 |
| | | 0 | 1 |
| 0 | 1 | 1 | 0 |
| | | 1 | 1 |
| 1 | 0 | 0 | 0 |
| | | 1 | 0 |
| 1 | 1 | 0 | 1 |
| | | 1 | 1 |

Виходячи з табл. 2.7 отримуємо:

- а) для переходу "0-0", $X = 0$, Y може дорівнювати 0 або 1;
- б) для переходу "0-1", $X = 1$, Y може дорівнювати 0 або 1;
- в) для переходу "1-0", X може дорівнювати 0 або 1, а $Y = 0$;
- г) для переходу "1-1", X може дорівнювати 0 або 1, а $Y = 1$.

Тоді матрицю переходів тригера можна записати у вигляді:

| | | |
|-------|-------|-------|
| 0 – 0 | 0 | b_1 |
| 0 – 1 | 1 | b_2 |
| 1 – 0 | b_3 | 0 |
| 1 – 1 | b_4 | 1 |

де:

b_1, b_2, b_3, b_4 – довільні сигнали (0 або 1).

Встановлення взаємозалежності між вхідними змінними тригера є обов'язковою умовою, що забезпечує можливість максимального спрощення схем з пам'яттю.

Тип тригера, який може бути використаний при реалізації 3D резюме за способом організації логічних зв'язків (із запуском (RS-тригери); з лічильним входом (T-тригери); тригери затримки (D-тригери); універсальні (JK-тригери); комбіновані (RST-, JKRS-, DRS-тригери)) або за способом запису інформації (синхронні і асинхронні). Тип тригеру обирається в залежності від задач 3D резюме. Наприклад, можуть бути тригери за способом запам'ятовування інформації. Якщо обираються тригери з логічною пам'яттю, то вони будуть виконані на логічних елементах І, АБО, НІ, І-НІ, АБО-НІ, І-АБО-НІ та інших, що є доволі зручно при програмуванні.

2.3 Евристичний алгоритм кодування станів автомату для 3D резюме

Перехід від абстрактного автомата до структурного і подальша його інтерпретація логічною схемою виконується шляхом кодування вхідних, вихідних сигналів і внутрішніх станів автомата наборами довічних змінних.

Для кодування вхідних сигналів виписується множина вхідних сигналів $X = \{x_1, x_2 \dots x_M\}$, які кодуються векторами довжини $K_{вх} = \text{int}(\log_2 M)$, де int – округлення до найближчого більшого цілого числа, M – кількість символів вхідного алфавіту. Кодування вхідних сигнал здійснюється довільно.

Вихідні сигнали автомата $Y = \{y_1, y_2, \dots y_S\}$ кодуються вектором довжини $K_{вих} = \text{int}(\log_2 S)$, де S – кількість символів вихідного алфавіту. Для кодування вихідних сигналів застосовуються вагові моделі алгоритмів [86, 87].

У даному випадку алгоритм наступний:

1) Кожному вихідному сигналу y_i співвідноситься ціле число P_i , яке дорівнює числу появ сигналу y_i в таблиці виходів автомата.

2) Числа P_i упорядковано зменшенням.

3) Вихідний сигнал y_i з найбільшою вагою ($P_i \text{ max}$) кодуються кодом, що містить всі нулі (00 ... 00);

4) Наступні L вихідних сигналів (де L – число розрядів в двійковому векторі вихідного сигналу) за списком убавання ваги кодуються кодами, що містять одну одиницю (00 ... 01, 00 ... 10, ..., 10 ... 00);

5) Для кодування наступних за списком убавання вихідних сигналів використовуються всі коди, що містять дві одиниці, потім три одиниці і т.д., поки не будуть закодовані всі вихідні сигнали.

В результаті буде отримане таке кодування, при якому чим частіше зустрічається вихідний сигнал в таблиці виходів, тим менше одиниць міститься в його коді.

Евристичний алгоритм кодування станів мінімізує сумарне число змін елементів пам'яті на всіх переходах автомата. Для 3D резюме подібний алгоритм призначений для того, щоб виключити можливість повтору питань.

Стан автомату $A = \{a_1, a_2, \dots, a_r\}$ кодується векторами довжини $K_{стан} = \text{int}(\log(r))$, де r потужність множини A (кількість станів). Відзначимо, що довжина вектору станів визначає і кількість елементів пам'яті (тригерів) даного автомата.

Стан автомату $A=\{a_1, a_2, a_3, a_4, a_5, a_6, a_7, a_8, a_9\}$ кодується векторами довжини $K_{стан}=\text{int}(\log(10))=4$. Для кодування використовується евристичний метод [88], який дозволяє найбільш ефективно добрати коди для отримання найкращого коефіцієнту ефективності кодування:

$$K_{ef}=W/P, \quad (2.4)$$

де:

P – загальна кількість переходів автомату, а W – вагова функція.

Вагова функція обчислюється за:

$$W=\sum t_{ij}, \quad (2.5)$$

де:

t_{ij} – відстань Хеммінга між кодами станів a_i і a_j .

Евристику використовують в системах штучного інтелекту під час пошуку простору рішень. Ця дія може бути виконана за допомогою функції, що вводиться у систему, яка дозволяє перебирати рішення за гілками, що ведуть до цільового вузла [89].

Суть алгоритму може бути описана наступними кроками:

1) Будується матриця M , що утримує всі пари номерів (a_r, b_r) переходів автомата.

2) Здійснюється ротація рядків матриці з умовою, що кожен наступний рядок утримує мінімум один елемент з попередніх рядків.

3) Кодуються стани першого рядка матриці M : $Ka_1=00\dots00$, $Kb_1=00\dots01$.

4) Викреслюється з матриці M перший рядок з закодованими станами.

Буде отримано матрицю M' .

5) В силу п. 3 в початковому рядку матриці M' буде закодований один елемент. Вибирається з першого рядка матриці M' незакодований елемент і позначається γ .

6) Будується матриця M_γ , на основі вибирання з матриці M' рядків, що містять γ . Нехай $B_\gamma = \{ \gamma_1, \gamma_2, \dots, \gamma_f, \dots, \gamma_F \}$ – множина елементів з матриці M_γ , які вже закодовані. Їх коди позначити через $K_{\gamma_1}, K_{\gamma_2}, \dots, K_{\gamma_f}, \dots, K_{\gamma_F}$ відповідно.

7) Для кожного K_{γ_f} ($f=1, 2, \dots, F$) знаходиться $C_{\gamma_f}^1$ – множина кодів, віддалених від K_{γ_f} на відстань Хеммінга, яка не зайнята у кодуванні станів автомату і дорівнює 1. Будується множина $D_\gamma^1 = \bigcup_{f=1}^F C_{\gamma_f}^1$. Якщо $D_\gamma^1 = 0$, то

будується нова множина $D_\gamma^2 = \bigcup_{f=1}^F C_{\gamma_f}^2$, де $C_{\gamma_f}^2$ – множина кодів, у яких кодова відстань з кодом K_{γ_f} дорівнює 2. Якщо $D_\gamma^2 = 0$, будується D_γ^3 і т.д., доки не буде знайдено $D_\gamma^n \neq 0$. Приймається, що $D_\gamma^n = \{K_{\gamma_1}, \dots, K_{\gamma_g}, \dots, K_{\gamma_G}\}$.

8) Для кожного K_{γ_g} знаходиться $w_{gf} = |K_{\gamma_g} - K_{\gamma_f}|^2$ – відстань Хеммінга між K_{γ_g} і всіма використовуваними кодами K_{γ_f} ($f=1, 2, \dots, F$).

9) Знаходиться $w_g = \sum_{f=1}^F w_{gf}$, ($g=1, \dots, G$).

10) З D_γ^n вибирається K_γ , у котрого $w_g = w_{g \min}$. Елемент γ кодується, як K_γ .

11) З матриці M' викреслюються рядки, в котрих обидва елемента закодовані, в результаті чого отримується нова матриця, яка також позначається M' . Якщо в матриці M' не залишилося жодного рядка, здійснюється перехід до п. 12. Якщо ця умова не виконується, то перехід здійснюється до п. 5.

12) Обчислюється функція (2.5): $w = \sum t_{ms}$, де $t_{ms} = |K_m - K_s|^2$.

13) Кінець.

Очевидно, що за наведеним алгоритмом згідно (2.4): $K \geq 1$, причому, чим менше значення K , тим краще результат кодування.

Наведений алгоритм не мінімізує переходи, а лише запускає процес переходу між питаннями 3D резюме в залежності від наданої відповіді. Для мінімізації переходів можна використати наступні бази:

1) Елементи I, АБО, НІ. Кількість входів елементів I, АБО – від 2 до 4-х. Здатність навантаження всіх елементів – до 10.

2) Елементи I-НІ. Кількість входів – від 2 до 6. Здатність навантаження – до 10.

3) Елементи АБО-НІ. Кількість входів – від 2 до 6. Здатність навантаження – до 10.

Для початку кодування слід побудувати матрицю переходів автомата M . Перші два рядки в матриці M кодуються: $K1 = 0000$ і $K5 = 0001$. Далі застосовується евристичний метод кодування, хід якого викладено у Додатку Б. Результат застосування евристичного методу кодування станів представлений в табл. 2.8.

Таблиця 2.8 – Результат кодування станів

| $A \backslash \alpha$ | α_1 | α_2 | α_3 | α_4 |
|-----------------------|------------|------------|------------|------------|
| a_1 | 0 | 0 | 0 | 0 |
| a_2 | 0 | 1 | 0 | 0 |
| a_3 | 0 | 1 | 0 | 1 |
| a_4 | 0 | 1 | 1 | 0 |
| a_5 | 0 | 0 | 0 | 1 |
| a_6 | 0 | 1 | 1 | 1 |
| a_7 | 0 | 0 | 1 | 0 |
| a_8 | 0 | 0 | 1 | 1 |
| a_9 | 1 | 0 | 1 | 0 |

Обчислимо коефіцієнт ефективності кодування станів $K=W/P$:

$$\begin{aligned}
 W = & \left| \begin{array}{c} 0000 \\ 0000 \end{array} \right| + \left| \begin{array}{c} 0000 \\ 0001 \end{array} \right| + \left| \begin{array}{c} 0000 \\ 0010 \end{array} \right| + \left| \begin{array}{c} 0100 \\ 0101 \end{array} \right| + \left| \begin{array}{c} 0100 \\ 0000 \end{array} \right| + \left| \begin{array}{c} 0100 \\ 0110 \end{array} \right| + \left| \begin{array}{c} 0101 \\ 0111 \end{array} \right| + \\
 & + \left| \begin{array}{c} 0101 \\ 0101 \end{array} \right| + \left| \begin{array}{c} 0101 \\ 0001 \end{array} \right| + \left| \begin{array}{c} 0110 \\ 0110 \end{array} \right| + \left| \begin{array}{c} 0110 \\ 0111 \end{array} \right| + \left| \begin{array}{c} 0110 \\ 0100 \end{array} \right| + \left| \begin{array}{c} 0001 \\ 0010 \end{array} \right| + \left| \begin{array}{c} 0001 \\ 0000 \end{array} \right| + \\
 & + \left| \begin{array}{c} 0001 \\ 0011 \end{array} \right| + \left| \begin{array}{c} 0111 \\ 0001 \end{array} \right| + \left| \begin{array}{c} 0111 \\ 1010 \end{array} \right| + \left| \begin{array}{c} 0111 \\ 0011 \end{array} \right| + \left| \begin{array}{c} 0010 \\ 0110 \end{array} \right| + \left| \begin{array}{c} 0010 \\ 0011 \end{array} \right| + \left| \begin{array}{c} 0010 \\ 0000 \end{array} \right| + \\
 & + \left| \begin{array}{c} 0011 \\ 0011 \end{array} \right| + \left| \begin{array}{c} 0011 \\ 0011 \end{array} \right| + \left| \begin{array}{c} 0011 \\ 0011 \end{array} \right| + \left| \begin{array}{c} 1010 \\ 1010 \end{array} \right| + \left| \begin{array}{c} 1010 \\ 1010 \end{array} \right| + \left| \begin{array}{c} 1010 \\ 1010 \end{array} \right| = 31
 \end{aligned}$$

| 1010 | | 0000 | | 0101 | | 0010 | | 1010 | | 0000 |

$$W=31; P=27;$$

$$K=31/27=1,148.$$

З урахуванням прийнятого кодування будується таблиця переходів і виходів структурного автомата (табл. 2.9), де Z – вхідні сигнали, Y – вихідні сигнали.

Таблиця 2.9 – Таблиця переходів і виходів структурного автомата

| Y | | w ₁ w ₂ | | | | | | | | |
|----|-------------------------------|--|------|------|------|------|------|------|------|------|
| | | 00 | 01 | 11 | 10 | 00 | 01 | 10 | 01 | 00 |
| x | A | $\alpha_1, \alpha_2, \alpha_3, \alpha_4$ | | | | | | | | |
| | z ₁ z ₂ | 0000 | 0100 | 0101 | 0110 | 0001 | 0111 | 0010 | 0011 | 1010 |
| 00 | | 0000 | 0101 | 0111 | 0110 | 0010 | 0001 | 0110 | 1010 | 0010 |
| 01 | | 0001 | 0000 | 0101 | 0111 | 0000 | 1010 | 0011 | 0000 | 1010 |
| 10 | | 0010 | 0110 | 0001 | 0100 | 0011 | 0011 | 0000 | 0101 | 0000 |

Структурна схема даного автомата представлена на рис. 2.2.

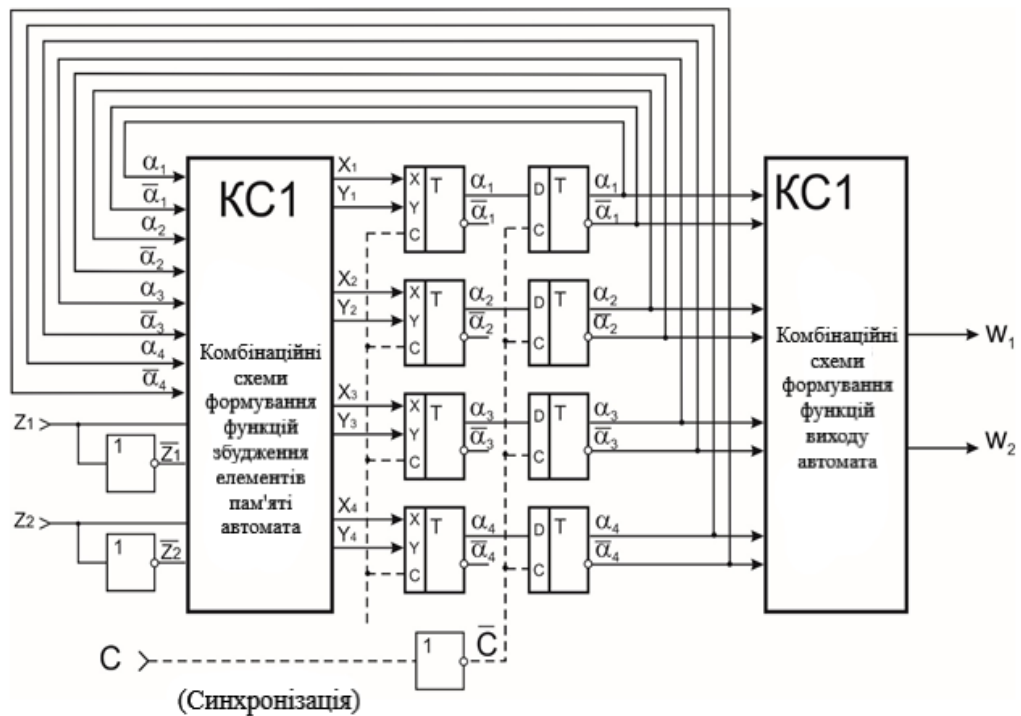


Рисунок 2.2 – Структурна схема автомата Мура

Позначення рис. 2.2:

X_i, Y_i – функції збудження елементів пам'яті автомата;

w_1, w_2 – функції виходів автомата;

α_i – функції зворотного зв'язку від елементів пам'яті до комбінаційної схемою.

У підсумку можна представити систему канонічних рівнянь даного автомата:

$$X_1 = \varphi_1(z_1, z_2, \alpha_1, \alpha_2, \alpha_3, \alpha_4)$$

$$Y_1 = \varphi_2(z_1, z_2, \alpha_1, \alpha_2, \alpha_3, \alpha_4)$$

$$X_2 = \varphi_1(z_1, z_2, \alpha_1, \alpha_2, \alpha_3, \alpha_4)$$

$$Y_2 = \varphi_2(z_1, z_2, \alpha_1, \alpha_2, \alpha_3, \alpha_4)$$

$$X_3 = \varphi_1(z_1, z_2, \alpha_1, \alpha_2, \alpha_3, \alpha_4)$$

$$Y_3 = \varphi_2(z_1, z_2, \alpha_1, \alpha_2, \alpha_3, \alpha_4)$$

$$X_4 = \varphi_1(z_1, z_2, \alpha_1, \alpha_2, \alpha_3, \alpha_4)$$

$$Y_4 = \varphi_2(z_1, z_2, \alpha_1, \alpha_2, \alpha_3, \alpha_4)$$

$$w_1 = w_1(\alpha_1, \alpha_2, \alpha_3, \alpha_4)$$

$$w_2 = w_2(\alpha_1, \alpha_2, \alpha_3, \alpha_4).$$

Після кодування алфавітів автомата і вибору елементів пам'яті, структурний синтез зводиться до синтезу комбінаційної схем, що реалізують систему канонічних рівнянь.

2.4 Формування функцій перемикавання та функцій виходу автомата

Після вибору елементів пам'яті та кодування вхідних сигналів і внутрішніх станів автомата структурний синтез зводиться до синтезу комбінаційної схеми, реалізує функції перемикавання.

Для автомата Мура зазначене можна представити наступним чином:

$$w_n = w_n(\alpha_1, \alpha_2, \dots, \alpha_k), \quad (2.6)$$

$$\varphi_i = \varphi_i(\alpha_1, \alpha_2, \dots, \alpha_k, z_1, z_2, \dots, z_r),$$

де:

w_n – функція виходів автомата;

φ_i – функція збудження елементів пам'яті;

α – функція зворотного зв'язку від елементів пам'яті до комбінаційної схеми;

z – кодовані вхідні сигнали.

А, наприклад, для автомата Мілі процес (2.6) буде представлений наступною схемою:

$$w_n = w_n(\alpha_1, \alpha_2, \dots, \alpha_k, z_1, z_2, \dots, z_r),$$

$$\varphi_i = \varphi_i(\alpha_1, \alpha_2, \dots, \alpha_k, z_1, z_2, \dots, z_r).$$

Щоб отримати функції виходів і збудження елементів пам'яті за вихідними таблицями переходів і виходів з урахуванням кодування вхідних, вихідних сигналів і станів автомата слід побудувати кодовані таблиці переходів і виходів. На прикладі автомата Мура, заданого таблицею переходів і виходів (табл. 2.10), можна виконати відповідне кодування автомата.

Таблиця 2.10 – Таблиця переходів і виходів для автомата Мура

| | у ₁ | у ₃ | у ₂ | у ₃ | у ₄ | у ₂ | у ₁ |
|----------------|----------------|----------------|----------------|----------------|----------------|----------------|----------------|
| А | а ₁ | а ₂ | а ₃ | а ₄ | а ₅ | а ₆ | а ₇ |
| х ₁ | 2 | 3 | 6 | 2 | 3 | 2 | 7 |
| х ₂ | 4 | 7 | 5 | 4 | 6 | 4 | 1 |

Для цього можна застосувати такі методи: для вхідних – довільний, для вихідних сигналів і станів – ваговий. Результати кодування представлені в таблиці 2.11. Довжина векторів кодів всіх алфавітів автомата складає:

$$K_{вх} = \text{int } \log_2 2 = 1, K_{вих} = \text{int } \log_2 4 = 2, K_{сост} = \text{int } \log_2 7 = 3.$$

Таблиця 2.11 – Результати кодування переходів і виходів

| X | z |
|----------------|---|
| x ₁ | 0 |
| x ₂ | 1 |

| Y | w ₁ | w ₂ | P |
|----------------|----------------|----------------|---|
| y ₁ | 0 | 0 | 2 |
| y ₂ | 0 | 1 | 2 |
| y ₃ | 1 | 0 | 2 |
| y ₄ | 1 | 1 | 1 |

а) Кодовані вхідні сигнали

| A | a ₁ | a ₂ | a ₃ | a ₄ | a ₅ | a ₆ | a ₇ |
|---|----------------|----------------|----------------|----------------|----------------|----------------|----------------|
| P | 1 | 3 | 2 | 3 | 1 | 2 | 2 |

б) Виходи автомату

| | α_1 | α_2 | α_3 | P |
|----------------|------------|------------|------------|---|
| a ₂ | 0 | 0 | 0 | 3 |
| a ₄ | 0 | 0 | 1 | 3 |
| a ₃ | 0 | 1 | 0 | 2 |
| a ₆ | 1 | 0 | 0 | 2 |
| a ₇ | 1 | 1 | 0 | 2 |
| a ₁ | 1 | 0 | 1 | 1 |
| a ₅ | 0 | 1 | 1 | 1 |

в) Кодування переходів автомату

г) Переходи загального зв'язку від елементів пам'яті

З урахуванням прийнятого кодування будемо таблицю переходів і виходів структурного автомата (табл. 2.12).

Таблиця 2.12 – Таблиця переходів і виходів структурного автомата

| Y | w ₁ w ₂ | | | | | | |
|---|-------------------------------|----|----|----|----|----|----|
| | 00 | 10 | 01 | 10 | 11 | 01 | 00 |
| A | $\alpha_1 \alpha_2 \alpha_3$ | | | | | | |

| | | | | | | | |
|---|-----|-----|-----|-----|-----|-----|-----|
| z | 101 | 000 | 010 | 001 | 011 | 100 | 110 |
| 0 | 000 | 010 | 100 | 000 | 010 | 000 | 110 |
| 1 | 001 | 110 | 011 | 001 | 100 | 001 | 101 |

Для побудови таблиці формування функцій збудження елементів пам'яті необхідно вибрати тип запам'ятовуючого елемента – тригера. У даному випадку вибирається Т-тригер з лічильним входом, який дозволить для реалізації задач 3D резюме здійснювати декілька спроб відповіді на одне питання із запам'ятовуванням попередньої відповіді та відслідковуванням прогресу навчання.

Як приклад можна розглянути матрицю переходів тригера, заданого скороченою таблицею переходів (табл. 2.4).

Від табл. 2.12, використовуючи матрицю переходів тригера, переходимо до табл. 2.13, яка являє собою систему булевих функцій, записану в табличному вигляді.

Таблиця 2.13 – Таблиця збудження елементів пам'яті

| | Початковий стан | | | Вх. z | Стан переходу | | | T1 | | T2 | | T3 | | Y | |
|----------------|-----------------|------------|------------|-------|---------------|------------|------------|----------------|----------------|----------------|----------------|----------------|----------------|-------|-------|
| | α_1 | α_2 | α_3 | | α_1 | α_2 | α_3 | X_1 | Y_1 | X_2 | Y_2 | X_3 | Y_3 | w_1 | w_2 |
| a ₁ | 1 | 0 | 1 | 0 | 0 | 0 | 0 | b ₃ | 0 | 0 | b ₁ | b ₃ | 0 | 0 | 0 |
| | | | | 1 | 0 | 0 | 1 | b ₃ | 0 | 0 | b ₁ | b ₃ | 0 | | |
| a ₂ | 0 | 0 | 0 | 0 | 0 | 1 | 0 | 0 | b ₁ | 1 | b ₂ | 0 | b ₁ | 1 | 0 |
| | | | | 1 | 1 | 1 | 0 | 1 | b ₂ | 1 | b ₂ | 0 | b ₁ | | |
| a ₃ | 0 | 1 | 0 | 0 | 1 | 0 | 0 | 1 | b ₂ | b ₃ | 0 | 0 | b ₁ | 0 | 1 |
| | | | | 1 | 0 | 1 | 1 | 0 | b ₁ | b ₄ | 1 | 1 | b ₂ | | |
| a ₄ | 0 | 0 | 1 | 0 | 0 | 0 | 0 | 0 | b ₁ | 0 | b ₁ | b ₃ | 0 | 1 | 0 |
| | | | | 1 | 0 | 0 | 1 | 0 | b ₁ | 0 | b ₁ | b ₄ | 1 | | |
| a ₅ | 0 | 1 | 1 | 0 | 0 | 1 | 0 | 0 | b ₁ | b ₄ | 1 | b ₃ | 0 | 1 | 1 |
| | | | | 1 | 1 | 0 | 0 | 1 | b ₂ | b ₃ | 0 | b ₃ | 0 | | |
| a ₆ | 1 | 0 | 0 | 0 | 0 | 0 | 0 | b ₃ | 0 | 0 | b ₁ | 0 | b ₁ | 0 | 1 |
| | | | | 1 | 0 | 0 | 1 | b ₃ | 0 | 0 | b ₁ | 1 | b ₂ | | |
| a ₇ | 1 | 1 | 0 | 0 | 1 | 1 | 0 | b ₄ | 1 | b ₄ | 1 | 0 | b ₁ | 0 | 0 |
| | | | | 1 | 1 | 0 | 1 | b ₄ | 1 | b ₃ | 0 | 1 | b ₂ | | |

Примітка: у табл. 2.13 є $X_1, Y_1, X_2, Y_2, X_3, Y_3$ – функції збудження елементів пам'яті за входами X_1, \dots, Y_3 відповідно.

З таблиці 2.13 можна отримати канонічні рівняння. Для цього, враховуючи велику кількість сигналів, що вимагають до визначення, значення функцій збудження тригерів записуються у вигляді карт Карно [90]. Для мінімізації канонічних виразів застосовують правила мінімізації неповністю визначених булевих функцій [91].

Мінімізація представлена на рис. 2.3 – 2.10.

| | | | | | |
|------------|---|--------------------|----|----|----|
| | | $\alpha_2\alpha_3$ | | | |
| | | 00 | 01 | 11 | 10 |
| α_1 | 0 | 1 | 1 | 1 | 0 |
| | 1 | 0 | 0 | * | 0 |

$$w_1 = \bar{\alpha}_1\bar{\alpha}_2 \vee \bar{\alpha}_1\alpha_3$$

| | | | | | |
|------------|---|--------------------|----|----|----|
| | | $\alpha_2\alpha_3$ | | | |
| | | 00 | 01 | 11 | 10 |
| α_1 | 0 | 0 | 0 | 1 | 1 |
| | 1 | 1 | 0 | * | 0 |

$$w_2 = \bar{\alpha}_1\alpha_2 \vee \alpha_1\bar{\alpha}_2\bar{\alpha}_3$$

Рисунок 2.3 – Мінімізація функції w_1 Рисунок 2.4 – Мінімізація функції w_2

| | | | | | | |
|-------|------------|--------------------|----|----|----|----|
| | | $\alpha_2\alpha_3$ | | | | |
| | | 00 | 01 | 11 | 10 | |
| z_1 | α_1 | 00 | 0 | 0 | 0 | 1 |
| | | 01 | b3 | b3 | * | b4 |
| | | 11 | b3 | b3 | * | b4 |
| | | 10 | 1 | 0 | 1 | 0 |
| | | | | | | |

| | | | | | | |
|-------|------------|--------------------|----|----|----|----|
| | | $\alpha_2\alpha_3$ | | | | |
| | | 00 | 01 | 11 | 10 | |
| z_1 | α_1 | 00 | b1 | b1 | b2 | b2 |
| | | 01 | 0 | 0 | * | 1 |
| | | 11 | 0 | b1 | * | 1 |
| | | 10 | b2 | b1 | 0 | b1 |
| | | | | | | |

Рисунок 2.5 – Мінімізація функції X_1 Рисунок 2.6 – Мінімізація функції Y_1

| | | | | | | |
|-------|------------|--------------------|----|----|----|----|
| | | $\alpha_2\alpha_3$ | | | | |
| | | 00 | 01 | 11 | 10 | |
| z_1 | α_1 | 00 | 1 | 0 | b4 | b3 |
| | | 01 | 0 | 0 | * | b4 |
| | | 11 | 0 | 0 | * | b3 |
| | | 10 | 1 | 0 | b3 | b4 |
| | | | | | | |

| | | | | | | |
|-------|------------|--------------------|----|----|----|---|
| | | $\alpha_2\alpha_3$ | | | | |
| | | 00 | 01 | 11 | 10 | |
| z_1 | α_1 | 00 | b2 | b1 | 1 | 0 |
| | | 01 | b1 | b1 | * | 1 |
| | | 11 | b1 | b1 | * | 0 |
| | | 10 | b2 | b1 | 0 | 1 |
| | | | | | | |

Рисунок 2.7 – Мінімізація функції X_2 Рисунок 2.8 – Мінімізація функції Y_2

| | | | | | |
|-------|------------|--------------------|----|----|----|
| | | $\alpha_2\alpha_3$ | | | |
| | | 00 | 01 | 11 | 10 |
| z_1 | α_1 | | | | |
| | 00 | 0 | b3 | b3 | 0 |
| | 01 | 0 | b3 | * | 0 |
| | 11 | 1 | b3 | * | 1 |
| | 10 | 0 | b4 | b3 | 1 |

| | | | | | |
|-------|------------|--------------------|----|----|----|
| | | $\alpha_2\alpha_3$ | | | |
| | | 00 | 01 | 11 | 10 |
| z_1 | α_1 | | | | |
| | 00 | b1 | 0 | 0 | b1 |
| | 01 | b1 | 0 | * | b1 |
| | 11 | b2 | 0 | * | b2 |
| | 10 | b1 | 1 | 0 | b2 |

Рисунок 2.9 – Мінімізація функції X_3 Рисунок 2.10 – Мінімізація функції Y_3

Якщо реалізувати варіант до визначення: $b1=b2=b3=b4=0$, то канонічні вирази приймуть вигляд:

$$X_1 = z\bar{\alpha}_1\bar{\alpha}_2\bar{\alpha}_3 \vee z\alpha_2\alpha_3 \vee z\bar{\alpha}_1\alpha_2\bar{\alpha}_3$$

$$Y_1 = \alpha_1\alpha_2$$

$$X_2 = \bar{\alpha}_1\bar{\alpha}_2\bar{\alpha}_3$$

$$Y_2 = z\bar{\alpha}_1\alpha_2\bar{\alpha}_3 \vee z\alpha_2\alpha_3 \vee z\bar{\alpha}_1\alpha_2$$

$$X_3 = z\alpha_1\bar{\alpha}_3 \vee z\alpha_2\bar{\alpha}_3$$

$$Y_3 = z\bar{\alpha}_1\bar{\alpha}_2\alpha_3$$

У загальному випадку, при мінімізації необхідно розглянути всі можливі варіанти до визначенні b_i та вибрати оптимальний, тобто той, який дає мінімальні значення функцій. У разі використання стандартного тригера, немає необхідності в до визначенні b_i [92].

2.5 Побудова функціональної схеми автомата

Розглянемо загальний процес побудови функціональної схеми автомата на елементах І, АБО, НІ.

Задана система булевих функцій (функцій виходу і функцій збудження елементів пам'яті) аналізується на наявність спільних частин. За допомогою інверторів формуються інверсні значення входних сигналів (інверсні значення сигналів зворотного зв'язку з запам'ятовуючих елементів отримують з інверсних виходів тригерів). На елементах «І» реалізуються кон'юнкції, що входять до булеві функції. Якщо кількість змінних, що входять в кон'юнкцію, перевищує кількість входів елемента, використовується двоступенева схема включення (рис. 2.12). Загальні частини реалізуються на окремих елементах (рис. 2.13), тобто загальна частина замінюється новою умовною змінною.

У відповідності з виразом булевої функції сигнали з виходів елементів «І» подаються на входи елементів АБО для реалізації диз'юнкції. Як і в попередньому випадку, можуть використовуватися загальні частини, двоступеневі схеми. Якщо навантаження на елемент перевищує його здатність навантаження, використовуються дублюючі елементи. Після реалізації комбінаційної схеми проводиться синтез функціональної схеми автомата.

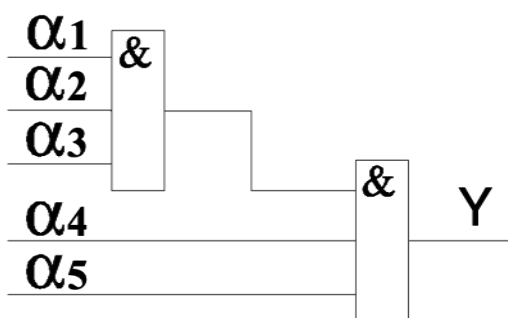


Рисунок 2.12 – Схема «І»

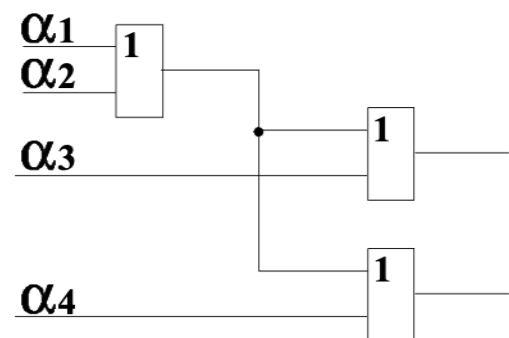


Рисунок 2.13 – Загальна частина

Основна ідея підходу до мінімізації внутрішніх станів автомата, який відповідає за логіку подачі завдань або питань в 3D резюме, полягає в такому

програмуванні резюме, при якому розбиття станів вихідного абстрактного автомата відбувається заміна кожного класу еквівалентності одним станом (за умови, що класи еквівалентних станів попарно не перетинаються). Це дозволить без дублювання проводити подачу питань або завдань резюме в залежності від попередньої відповіді респондента. Для цього приймаємо, що два стану автомата a_m і a_s називаються еквівалентними ($a_m \equiv a_s$), якщо $\lambda(a_m, X) = \lambda(a_s, X)$ для всіх можливих питань на вході з деякою заданою довжиною X . При цьому приймається, що у разі, коли a_m і a_s не еквівалентні, то ці два стани автомату є абсолютно різними. К-еквівалентність при цьому виступає більш слабкою. Стани a_m і a_s К-еквівалентні, якщо $\lambda(a_m, X_k) = \lambda(a_s, X_k)$ для всіх можливих вхідних слів довжини K . А якщо мова йде лише про автомат Мура, то застосовується поняття нульової еквівалентності та розбиття множин на «нуль»-класи. Тобто, це автомата Мура, які однаково відмічені вхідними сигналами.

В задачі створення 3D резюме вище було використано підхід із використанням автомату Мура. Для створення алгоритму мінімізації використовуємо табл. 2.14.

Таблиця 2.14 – Таблиця всіх переходів і виходів

| | | | | | | | | | | | | |
|----|-----------------|-----------------|----------------|-----------------|----------------|-----------------|----------------|-----------------|----------------|-----------------|-----------------|-----------------|
| У | у ₁ | у ₁ | у ₃ | у ₃ | у ₃ | у ₂ | у ₃ | у ₁ | у ₂ | у ₂ | у ₂ | у ₂ |
| А | а ₁ | а ₂ | а ₃ | а ₄ | а ₅ | а ₆ | а ₇ | а ₈ | а ₉ | а ₁₀ | а ₁₁ | а ₁₂ |
| х1 | а ₁₀ | а ₁₂ | а ₅ | а ₇ | а ₃ | а ₇ | а ₃ | а ₁₀ | а ₇ | а ₁ | а ₅ | а ₂ |
| х2 | а ₅ | а ₇ | а ₆ | а ₁₁ | а ₉ | а ₁₁ | а ₆ | а ₄ | а ₆ | а ₈ | а ₉ | а ₈ |

На прикладі табл. 2.14 виконуються наступні дії:

1) Розбиття π_0 :

$$\pi_0 = \{B_1, B_2, B_3\};$$

$$B_1 = \{a_1, a_2, a_8\}, B_2 = \{a_6, a_9, a_{10}, a_{11}, a_{12}\}, B_3 = \{a_3, a_4, a_5, a_7\}.$$

2) Побудуємо робочу табл. 2.15 розбиття π_0 .

Таблиця 2.15 – Робоча таблиця розбиття π_0

| У | В ₁ | | | В ₂ | | | | | В ₃ | | | |
|----|----------------|----------------|----------------|----------------|----------------|-----------------|-----------------|-----------------|----------------|----------------|----------------|----------------|
| А | а ₁ | а ₂ | а ₈ | а ₆ | а ₉ | а ₁₀ | а ₁₁ | а ₁₂ | а ₃ | а ₄ | а ₅ | а ₇ |
| х1 | В ₂ | В ₂ | В ₂ | В ₃ | В ₃ | В ₁ | В ₃ | В ₁ | В ₃ | В ₃ | В ₃ | В ₃ |
| х2 | В ₃ | В ₃ | В ₃ | В ₂ | В ₂ | В ₁ | В ₂ | В ₁ | В ₂ | В ₂ | В ₂ | В ₂ |

3) Виконаємо розбиття π_1 із занесенням у наступну робочу табл. 2.16:

$$\pi_1 = \{C_1, C_2, C_3, C_4\};$$

$$C_1 = \{a_1, a_2, a_8\}, C_2 = \{a_6, a_9, a_{11}\}, C_3 = \{a_{10}, a_{12}\}, C_4 = \{a_3, a_4, a_5, a_7\}.$$

Таблиця 2.16 – Робоча таблиця розбиття π_1

| У | С ₁ | | | С ₂ | | | С ₃ | | С ₄ | | | |
|----|----------------|----------------|----------------|----------------|----------------|-----------------|-----------------|-----------------|----------------|----------------|----------------|----------------|
| А | а ₁ | а ₂ | а ₈ | а ₆ | а ₉ | а ₁₁ | а ₁₀ | а ₁₂ | а ₃ | а ₄ | а ₅ | а ₇ |
| х1 | С ₃ | С ₃ | С ₃ | С ₄ | С ₄ | С ₄ | С ₁ | С ₁ | С ₄ | С ₄ | С ₄ | С ₄ |
| х2 | С ₄ | С ₄ | С ₄ | С ₂ | С ₂ | С ₂ | С ₁ | С ₁ | С ₂ | С ₂ | С ₂ | С ₂ |

4) Аналогічно виконується для π_2 :

$$\pi_2 = \{D_1, D_2, D_3, D_4\};$$

$$D_1 = \{a_1, a_2, a_8\}, D_2 = \{a_6, a_9, a_{11}\}, D_3 = \{a_{10}, a_{12}\}, D_4 = \{a_3, a_4, a_5, a_7\}.$$

5) З наведеного витікає, що розбиття π_2 є повторенням π_1 . У цьому випадку процедуру можна вважати завершеною.

6) Наявні класи еквівалентності D_1, D_2, D_3, D_4 . З цих класів вибирається по одному представнику, припустимо, за мінімальним номером: $A = \{a_1, a_3, a_6, a_{10}\}$.

7) З таблиці 2.12 вибираються стани, які не відповідають поставленій задачі (т. зв. «зайві»). Результат заноситься у окрему таблицю (табл. 2.17), яка і визначає мінімальний автомат Мура.

Таблиця 2.17 – Мінімальний автомат Мура

| | | | | |
|----|-----------------|----------------|----------------|-----------------|
| У | у ₁ | у ₃ | у ₂ | у ₂ |
| А | а ₁ | а ₃ | а ₆ | а ₁₀ |
| х1 | а ₁₀ | а ₃ | а ₃ | а ₁ |
| х2 | а ₃ | а ₆ | а ₆ | а ₁ |

Реалізацію алгоритму можна представити у вигляді концептуальної схеми 3D резюме (рис. 2.14).

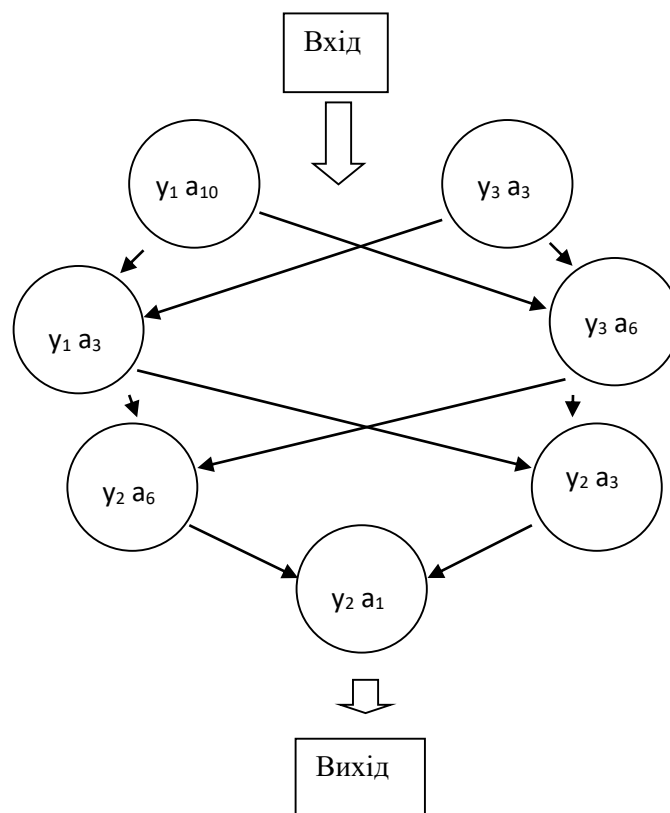


Рисунок 2.14 – Концептуальна схема реалізації 3D резюме

На рис. 2.14 можна побачити, що на вході 3D резюме відбувається довільний вибір між двома наборами питань для відбору персоналу. Далі питання перемішуються, бо обидва набори відповідають одному рівню підготовки персоналу. При цьому питання не дублюються. Перемішування питань відбуваються при переходах за умовою «коли попереднє завдання

випадку K -еквівалентні стани є еквівалентними і система зараховує відповідь респондента, як правильну.

2.6 Висновки за розділом 2

Другий розділ роботи дозволяє вивести наступні положення:

1) особливістю вирішеної у розділі 2 задачі є те, що запропонований підхід дозволяє реалізувати інструментальний засіб для опису дій людини, що проходить тестування за допомогою мережі Інтернет, коли на початку є деякі вхідні слова (тезаурус спеціальності, що закладений у тести, завдання, співбесіду у дистанційному форматі), що описують розуміння респондентом теми співбесіди, а на виході, після складного перебору обмежень, є сформований системою висновок про відповідність даного претендента наведеним професійним вимогам. Звичайно, остаточне рішення приймає людина (менеджер по персоналу), але основний відбір здійснює саме система, яка разом з респондентом переходить за циклами вибору та запам'ятовує питання і відповіді на них, співвідносячи це з конкретною зареєстрованою в системі особою;

2) використання абстрактних автоматів для реалізації дій повтору і вибору у алгоритмі моделювання інтелектуальної діяльності людини дозволяє здійснювати переходи між питаннями в залежності від попередньої відповіді респондента. Наприклад, якщо відповідь претендента при заповненні 3D резюме не співпадає хоча б з одним з ключових слів, то наступне завдання 3D резюме не повинне активуватися;

3) програмування тригерів, як схеми зі стійкими станами перебування, допоки на зміниться керуючий вплив, і які дозволяють перетворити та

запам'ятати інформацію, що отримала система, можливе за допомогою будь-яких сучасних мов програмування. Приклад програмування тригера буде наведений в практичному розділі роботи. Тип тригера, який може бути використаний при реалізації 3D резюме може бути обрано за способом організації логічних зв'язків, з лічильним входом, із затримкою, універсальні, комбіновані і, навіть, за способом запису інформації. Тип тригера обирається в залежності від задач 3D резюме;

4) наведений евристичний алгоритм кодування станів мінімізує сумарне число змін елементів пам'яті на всіх переходах автомата. Це дозволяє виключити можливість вибору одного питання із множини питань для того ж самого користувача, а також вибору одного набору питань для різних користувачів, що проходять відбір на роботу. Це виключає вплив людського фактору на автоматизовану систему і дозволяє отримати об'єктивні результати;

5) матеріали, що подані в розділі, опубліковані в [80, 96].

РОЗДІЛ 3

ЛОГІКО-ЙМОВІРНІСНИЙ ПІДХІД ДО ФОРМУВАННЯ МНОЖИНИ ЗАВДАНЬ ДЛЯ НЕЗАЛЕЖНИХ ВИПРОБУВАНЬ В СИСТЕМАХ ПІДБОРУ КАДРІВ

3.1 Генерування вибору випадкового питання для формування множини питань в системі тестування знань працівника

При переборі переходів від одного питання до іншого в системі відбору персоналу окрім задачі власне забезпечення цих переходів із запам'ятовуванням результатів входу і виходу, слід також реалізувати модель вибору завдань із множини / переліку всіх завдань, що закладені в систему.

При цьому завдання повинні поступати у порядку збільшення або зменшення складності для уточнення рівня підготовки працівника або кандидата на роботу. Завдання можуть бути з підказкою або без, із використанням тезаурусу професійного словника, забезпечувати деяку ймовірність проходження завдань (відсоток виконаних завдань для отримання задовільного результату). Завдання не повинні повторюватися в межах виконання одного тестування і не повинні мати однакової черговості відкриття при низці перевірок знань працівників.

Для реалізації логіко-ймовірнісної моделі вибору завдань із бази можна застосувати різні підходи обробки інформації, створюючи особливий комплекс – гібридизацію моделей для обробки інформації [97], які дозволяють поєднати елементи та засоби штучного інтелекту для вибору завдання [98], що дозволить у заданий часовий термін, шляхом логічного перебору питань, виявити здатність людини виконувати професійні дії та застосовувати певні навички.

В роботі [99] наведені методологічні основи подібного поєднання елементів штучного інтелекту із різноманітними підходами та методами для побудови прикладних систем. Для реалізації задачі дисертаційної роботи пропонується використати підхід Г.П. Донця, В.А. Пепеляєва та О.М. Трофимчука [100], зокрема, принцип оптимальності та Лему 2 про оптимальне розбиття на кількість груп, та застосувати стратегію, яка дозволяє з набору даних сформулювати вибірки питань, доповнюючи зазначений підхід.

Припускаємо, що маємо базу даних із декількох тисяч завдань різної складності, які за [100] оптимально розбиті на сукупності. Необхідно розробити імітаційну модель та алгоритм визначення випадкового питання, яке потрапляє до обох вибірок одночасно з найменшою долею ймовірності.

Для порівняння двох вибірок, сформованих з випадкових питань, можна скористатися наступною методикою:

а) розподіляємо обидві вибірки на 10 однакових частин з однаковими верхніми та нижніми межами – лівою і правою;

б) підраховуємо частоти попадання випадкових величин у кожний з інтервалів;

в) для встановлення однорідності вибірок використовуємо непараметричний критерій Пірсона [101]:

$$\chi^2 = n_1 n_2 \sum_{i=1}^{\ell} \frac{1}{m'_i + m''_i} \cdot \left(\frac{m'_i}{n_1} - \frac{m''_i}{n_2} \right)^2, \quad (3.1)$$

де:

n_1, n_2 – кількість елементів першої та другої вибірок;

ℓ – кількість груп, на яке розподілена кожна з вибірок;

m'_i – кількість елементів першої вибірки, що попали у i -й інтервал;

m''_i – кількість елементів другої вибірки, що попали у i -й варіант.

Цей критерій при достатньо великих n_1 та n_2 розподіляється по закону Пірсона « χ^2 квадрат» з $\ell - 1$ ступенем свободи. Тому процедура перевірки полягає у:

- 1) розрахунку по (3.1) сподіваного значення критерію;
- 2) знаходженню критичного значення χ^2 квадрат при заданому рівні значимості по таблицям або за допомогою комп'ютерних програм;

Якщо сподіване значення менше за критичне – немає підстав відкинути гіпотезу про однорідність розподілу. Якщо навпаки – розподіл у обох вибірках слід вважати різним.

Для тестового моделювання подібного підходу можна використати MS Excel, сформувавши таблиці вибірки та розраховавши показники відносних накопичених частот як прообраз інтегральної функції розподілу (табл. 3.1).

Таблиця 3.1 – Накопичені відносні частоти

| № | Інтервал | | <i>P</i> | № | Інтервал | | <i>P</i> |
|---|-----------|------------|----------|----|-----------|------------|----------|
| | Ліва межа | Права межа | | | Ліва межа | Права межа | |
| 1 | 0 | 11 | 0,3065 | 6 | 55 | 66 | 0,9885 |
| 2 | 11 | 22 | 0,6565 | 7 | 66 | 77 | 0,9955 |
| 3 | 22 | 33 | 0,8515 | 8 | 77 | 88 | 0,997 |
| 4 | 33 | 44 | 0,9325 | 9 | 88 | 99 | 0,9995 |
| 5 | 44 | 55 | 0,9695 | 10 | 99 | 110 | 1 |

Для моделювання однорідності вибірки нижче наведено лістинг додатку у Visual Basic for Applications (VBA) для тестування запропонованого підходу.

```
Sub Генератор_()
''
Randomize
'**** Початок циклу розрахунків ****
For k = 1 To 2000
'****Генерація випадкового номеру питання
a_rand = Rnd
'**** Вибір інтервалу за допомогою селектору***
Select Case a_rand
Case 0 To 0.3065
```

```

        in_1 = 1
Case 0.3065 To 0.6565
        in_1 = 2
Case 0.6565 To 0.8515
        in_1 = 3
Case 0.852 To 0.9325
        in_1 = 4
Case 0.9325 To 0.9695
        in_1 = 5
Case 0.9695 To 0.9885
        in_1 = 6
Case 0.9885 To 0.9955
        in_1 = 7
Case 0.9955 To 0.997
        in_1 = 8
Case 0.997 To 0.9995
        in_1 = 9
Case 0.9995 To 1
        in_1 = 10
End Select
'MsgBox a_rand & " " & in_1 ' Відладочний оператор
'***Генерація випадкового номеру питання всередині
інтервалу**
aa = 11 * (in_1 - 1) + 11 * Rnd
'*** Округлення до десятих
aa = Int(10 * aa) / 10
'*** Виведення результату ***
Cells(3 + k, 2) = aa
Next k
End Sub

```

Перевірити гіпотезу про однаковий розподіл можна за зазначеною формулою (3.1). Дані для абсолютних частот влучення у інтервали наведено у табл. 3.2. Для розрахунків приймаємо, що у нас є дві сформовані вибірки питань обсягом $n_1=n_2=2000$.

Таблиця 3.2 – Перевірка гіпотези про однаковий розподіл вибірки

| Інтервал | m_{1i} | m_{2i} | $\frac{1}{m_{1i} + m_{2i}}$ | $\left(\frac{m_{1i}}{n_1} + \frac{m_{2i}}{n_2}\right)^2$ | Добуток |
|-----------------|----------|----------|-----------------------------|--|------------|
| 1 | 613 | 597 | 0,000826 | 0,000064 | 5,2893E-08 |
| 2 | 700 | 699 | 0,000715 | 2,5E-07 | 1,787E-10 |
| 3 | 390 | 385 | 0,00129 | 6,25E-06 | 8,0645E-09 |
| 4 | 162 | 178 | 0,002941 | 0,000064 | 1,8824E-07 |
| 5 | 74 | 72 | 0,006849 | 0,000001 | 6,8493E-09 |
| 6 | 38 | 46 | 0,011905 | 0,000016 | 1,9048E-07 |
| 7 | 14 | 8 | 0,045455 | 0,000009 | 4,0909E-07 |
| 8 | 3 | 3 | 0,166667 | 0 | 0 |
| 9 | 5 | 4 | 0,111111 | 2,5E-07 | 2,7778E-08 |
| 10 | 1 | 1 | 0,5 | 0 | 0 |
| | | | | Сума | 8,8357E-07 |
| Ступінь свободи | Висновок | | | | |
| 9 | 3,5343 | 16,9 | 0,05 | Різниця не значима | |

Як впливає з результатів табл. 3.2, можна вважати, що програма-генератор генерує випадкові числа за тім ж самим законом, що наявний у заданої вибірці.

3.2 Генерація випадкового номера завдання за заданої аналітичної функції розподілу

Випадковий номер завдання із бази можна обирати за заданої аналітичної функції розподілу. Наприклад, приймається, що задана густина розподілу на відрізьку $[0; 2]$. Тобто:

$$f(x) = A \cdot \left[1 - (x - 1)^{\frac{2}{3}}\right]; \quad x \in [0; 2]. \quad (3.2)$$

Далі розраховується інтегральна функція розподілу для заданого інтервалу, та отримується диференціальна функція. А потім можна

застосувати наступний алгоритм генерації випадкових чисел методом зворотних функцій:

- 1) генеруємо рівномірно розподілене випадкове число $R \in (0,1)$;
- 2) знаходимо необхідне випадкове число X , як рішення рівняння:

$$F(X) = R, \quad (3.3)$$

де:

$F(X)$ – задана інтегральна функція розподілу.

У прикладі, що розглядається, слід знайти рішення рівняння:

$$\phi(x) = \frac{5}{4} \cdot \left[x - \frac{3}{5} (x - 1)^{\frac{5}{3}} - \frac{3}{5} \right] - R = 0, \quad (3.4)$$

причому, з області визначення перемінної $x \in [0;2]$.

Вважаючи, що функція (3.4) є складною і може мати корені, що не належать області задачі, для запобігання знаходження «фіктивного» кореню доцільно використовувати метод дихотомії (половинного ділення), який не виводить за межі $x \in [0; 2]$. Алгоритм цього методу наступний:

- 1) задаємо ліву (a) та праву (b) межі області визначення виразів, потрібну точність результату (*eps*);
- 2) Перевіряємо, чи виконується нерівності $f(a) < 0$, $f(b) > 0$. Якщо виконуються протилежні нерівності – міняємо місцями a та b ;
- 3) розраховуємо:

$$c = \frac{a+b}{2}; \quad (3.5)$$

- 4) перевіряємо виконання умови:

$$|b-a| < \epsilon. \quad (3.6)$$

Якщо (3.6) виконується – отримане рішення, кінець алгоритму.

5) якщо (3.6) не виконується – перевіряємо умову:

$$f(c) > 0. \quad (3.7)$$

б) у разі, коли (3.7) виконується – позначаємо $b = c$ та переходимо до кроку 3;

7) коли (3.7) не виконується – позначаємо $a = c$ і переходимо до кроку 3, виконуючи алгоритм до завершення.

Лістинг програми генерації випадкового номеру питання на мові VBA, наведено нижче:

```
Option Explicit
Dim r, a, b, c As Double
Dim i As Integer
Sub Equatio_2()

Randomize
For i = 1 To 2000
    r = Rnd

    a = 0
    b = 2
    If FR(a, r) > 0 And FR(b, r) < 0 Then
        c = a
        a = b
        b = c
    End If
    While Abs(b - a) > 0.00000001
        c = (a + b) / 2
        If FR(c, r) > 0 Then
            b = c
```

```

Else
    a = c
End If
Wend
'MsgBox c & " " & FR(c, r)
Cells(3 + i, 2) = c
Next i
'
End Sub
Function FR(x, r)
    FR = 5/4*(x-0.6*Sgn(x-1)*Abs(x-1)^(5/3)-0.6)-r
End Function

```

Генерація подібної вибірки можлива також і за методом Неймана-Пірсона, який дозволяє мінімізувати помилки обчислень [102]. У цьому випадку пропонується реалізація наступного алгоритму генерації:

- 1) генеруються 2 рівномірно розподілених випадкових числа R_1 і R_2 .
- 2) на їх основі розраховуються 2 випадкових числа:

$$y_1 = a + (b - a) \cdot R_1, \quad (3.8)$$

$$y_2 = f_{max} \cdot R_2, \quad (3.9)$$

де:

f_{max} – максимум диференціальної функції розподілу (густини розподілу);

- 3) перевіряється, чи виконується нерівність:

$$y_2 \leq f(y_1); \quad (3.10)$$

- 4) якщо нерівність (3.10) виконується – запам'ятовується число y_1 , яке розподілено за заданим законом;

- 5) якщо нерівність (3.10) не виконується – розрахунки повторюються, починаючи з кроку 1.

Нижче наведено варіант лістингу програми на мові VBA, за наведеним алгоритмом.

```

Sub Neumann()
Randomize
fmax = ff(1)
For i = 1 To 2000
    y2 = 1000
    While y2 > ff(y1)
        r1 = Rnd
        r2 = Rnd
        y1 = 2 * r1
        y2 = r2 * fmax
    Wend

    Cells(3 + i, 2) = y1
Next i

,

End Sub
Function ff(z)
ff = 1.25 * (1 - Abs(z - 1) ^ (2 / 3))
End Function

```

Для тестування алгоритмів, наведених у цьому підрозділі роботи, була використана та ж сама генерована вибірка з 2-х тисяч номерів питань. Розмах вибірки – від 0 до 2000, розподіл – на 10 частин довжиною 200 (0,2). Результати наведено у таблиці 3.3.

Таблиця 3.3 – Результати тестування алгоритмів генерації випадкового номера завдання за заданої аналітичної функції розподілу

| Інтервал | | Вибірки | | $\frac{1}{m'_i + m''_i} \cdot \left(\frac{m'_i}{n_1} - \frac{m''_i}{n_2} \right)^2$ |
|---------------|------------|------------------|-----------------------|--|
| Ліва межа | Права межа | Обернена функція | Метод Неймана-Пірсона | |
| 0 | 0,2 | 41 | 29 | 5,14286E-07 |
| 0,2 | 0,4 | 97 | 116 | 4,23709E-07 |
| 0,4 | 0,6 | 200 | 187 | 1,09173E-07 |
| 0,6 | 0,8 | 308 | 247 | 1,67613E-06 |
| 0,8 | 1 | 372 | 370 | 1,34771E-09 |
| 1 | 1,2 | 416 | 422 | 1,07399E-08 |
| 1,2 | 1,4 | 266 | 290 | 2,58993E-07 |
| 1,4 | 1,6 | 178 | 198 | 2,65957E-07 |
| 1,6 | 1,8 | 82 | 97 | 3,14246E-07 |
| 1,8 | 2 | 40 | 44 | 4,7619E-08 |
| | | | сума | 3,6222E-06 |
| $\chi^2_{сп}$ | α | k | $\chi^2_{кр}$ | Висновок |
| 14,48879 | 0,05 | 9 | 16,91896 | Вибірки однорідні |

Як випливає з результатів таблиці 3.3, обидві отримані вибірки є однорідними, тобто, обидва методи дають статистично не відмінні вибірки. Для реалізації алгоритму вибору завдання з наявної бази за номерами завдання можна використати будь-який із зазначених методів.

3.3 Логічний підхід до вирішення задачі відповідності отриманої відповіді поставленому питанню

3.3.1 Синтаксис обчислення висловлювань

Для прийняття рішень за умов невизначеності з відбору кадрів необхідно, щоб подібний інструментальний засіб міг вирішувати задачу відповідності отриманої відповіді поставленому питанню. Тільки після цього, оцінюючи виконання попереднього завдання, для респондента може бути відкрите наступне завдання.

Задачу відповідності відповіді можна вирішити за допомогою логіки висловлювань, зокрема, за допомогою обчислювання висловлювань або логічного програмування [103].

Словник обчислювань висловлювань може допомогти у формуванні висновку відносно нових висловлювань, отриманих від респондента. Правила побудови описують ті вирази, які є об'єктами мови, або формулами мови. Тобто, вислів є складеним висловлюванням, яке побудоване за певними мовними правилами.

Сукупність правил може бути представлена наступним чином:

а) базисом є слово або висловлювання, яке входить до тезаурусу сфери опитування і визнається формулою;

б) індукційним кроком є припущення того, що деякі ключові слова X та Y у відповіді респондента є формулами, у такому випадку формулами є деякі логічні висловлювання: кон'юнкція, диз'юнкція, логічне заперечення, імплікація, еквівалентність;

в) обмеженням правил є те, що застосовуються лише базис та індукційний крок для визначення істини або хибі.

Фактично застосування такого правила є рух по вузлам деякого дерева висловлювань: кореневий вузол – початкове слово, а далі вузли-зв'язки покрокові, які дозволяють на кожному індукційному кроці виявити відповідність відповіді поставленій задачі.

Зазначене можна представити за підходом [104] формулою:

$$(p \wedge (q \vee r)), \quad (3.11)$$

де позначення p , q , r – є різноманітними можливими висловлюваннями за завданням тесту. Індукційний крок у (3.11) застосовується двічі – при побудові спочатку висловлювання $(q \vee r)$, а потім поєднання з висловлюванням p .

Якщо при побудові формули X формула Y виступить елементом індукції (тобто, істинно, але не \wedge доведено), то у цьому випадку Y є підформулою.

Подібні зв'язки можна продемонструвати за допомогою деревовидної форми (рис. 3.1).

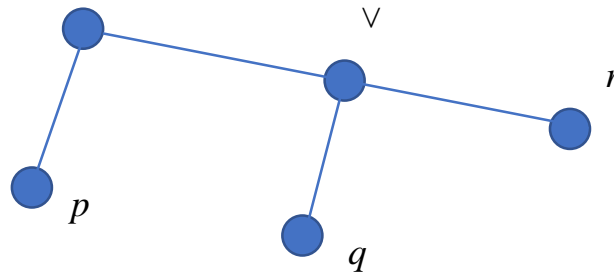


Рисунок 3.1 – Деревовидна форма формули (3.11)

Множина подібних формул підмножиною множини всіх слів та їх поєднань, які побудовані з вхідних символів, слів, словосполучень. Ця множина слів є скінченною, такою, яку можна перерахувати та відповідною тезаурусу за темою опитування (темою завдань). Множина формул є рекурсивною, тобто, можна визначити, чи є слово формулою, чи навпаки. Таким чином відбувається диференціювання формул за критерієм істини/хиби.

3.3.2 Алгоритм знаходження відповідності отриманої відповіді за словами-еталонами та ключовим словом завдання

При використанні ШІ так само, як і у випадку роботи з людиною, існує базовий словник, в який додаються необхідні базові терміни, на основі яких формуються логічні ланцюги з формули та індукційних кроків. Слова-еталони, які може використати респондент при відповіді на завдання, слід порівняти з ключовим словом, яке виступає базисом вірної відповіді на завдання.

Базовий словник системи повинен забезпечувати дві вимоги: слова-еталони повинні бути упорядковані, а сам словник повинен мати інформаційну надлишковість. Виникнення нового слова відповіді, яке за індукційного кроку не суперечить обмеженням, додається до базового словника.

Приймаємо, що:

– $A_j = (a_1 \dots a_i \dots a_n)$ - j -е слово базового словника системи;

– $i = \overline{1, n}; j = \overline{1, N}$;

– q – базові терміни та слова-еталони.

Кожному терміну та слову-еталону a_i співвідносимо деяке число $\alpha_{k_i} = \{\alpha_{0_i} \dots \alpha_{k_i} \dots \alpha_{(q-1)_i}\}$, за умови: $\alpha_0 = 0$; $\alpha_{q-1} = q - 1$; $\alpha_{k+1} = \alpha_k + 1$. Тобто, присвоюємо порядковий номер q з відповідністю базового терміну та слів еталонів окремому завданню з множини завдань системи. Словник буде вважатися упорядкованим при виконанні:

$$\sum_{i=1}^n (\alpha_{k_i} \cdot q^{n-i})_j \succ \sum_{i=1}^n (\alpha_{k_i} \cdot q^{n-i})_{j+1}. \quad (3.12)$$

Формування множини слів-еталонів за (3.12) упорядковуються за убутанням значення $\frac{\alpha_{k_i}}{\alpha_{k_{i+1}}} = q$. Тобто, останнє слово-еталон вказує на відповідь респондента, яка є задовільною і припускає відкриття подальшого кроку в проходженні тестування знань та вмінь.

Інформаційна надлишковість словника системи повинна забезпечувати вимогу, за якою з q^n різноманітних значень комбінацій слів n для вірної відповіді на завдання використовується тільки $N!$ комбінацій, які є лише частиною словника системи q^n ($q^n \gg N$). У цьому випадку створюється можливість «навчання» системи перебирати наявні у словнику слова та створювати за індукційними кроками нові конструкції можливих варіантів відповіді.

Алгоритм знаходження відповідності отриманої відповіді за словами-еталонами та ключовим словом завдання може бути наступним:

- 1) початок алгоритму: респондент ввів відповідь: a_1, a_2, \dots, a_i ;
- 2) формується поточна множина m_i відповідних термінів та слів-еталонів із базового словника: $a_1, a_1 a_2, a_1 a_2 a_3, \dots, a_1 a_2 \dots a_i$.

Варто зазначити, що при вводі слів відповіді a_1, a_2, \dots , з врахуванням упорядкованості термінів та слів-еталонів, область пошуку зменшується $m_1 \rangle m_2 \rangle \dots \rangle m_i$. Ідеальний випадок створення базового словника – $m_i/m_{i+1} \approx q$, тобто наявність випадкового характеру розподілу базових термінів та слів-еталонів серед q^n можливих поєднань термінів відповіді $a_1 \dots a_n$;

3) при введенні слова відповіді a_i з поточного довідника m_i для ідентифікації відповідності надається множина обсягом m термінів та слів-еталонів;

4) якщо шукана відповідність у наданій множині m відсутня, то відбувається пошук відповідності з меншою ймовірністю, що дозволяє звужувати область пошуку до множини m_{i+1} . Процес пошуку відповідності відбувається до знаходження шуканого терміну;

5) якщо відповідність не знайдено – до розгляду приймається наступне слово;

б) алгоритм завершено, коли всі слова відповіді проаналізовані.

У підсумку отримується відповідь за виявленими відповідностями, при чому $m_p \leq m$, або відповідь дорівнює «нуль», при незнаходженні відповідності.

3.4 Використання правил логіки при формуванні базового словника системи тестування знань і навичок персоналу

3.4.1. Модель процесу обчислення висловлювань

В системі тестування знань персоналу наявне деяке питання, відповідь якого описана формулою Y . При застосування індукційного кроку за правилами отримане висловлювання Y_i , яке може бути відповідним відповіді респондента, а може і не бути таким.

У випадку хибної відповіді та невідповідності з висловлюванням Y_i , можна отримати висловлювання Y_{i+1} , яке не є абсолютно точним, але дозволяє співставити відповідь респондента з отриманням оцінки знань, що і є метою створеної системи. У випадку відповідності відповіді респондента висловлюванню Y_{i+1} , процес завершується. Ймовірність проходження тесту з висловлюванням Y_i дорівнює P_i . Ця ймовірність не залежить від ймовірностей результатів з іншими висловлюваннями $i - 1, i - 2, \dots$

Тестування проводиться багаторазово з однаковою послідовністю використовуваних висловлювань $i = 1, 2, \dots, n$. Якщо серія з n опитувань закінчилася без знаходження відповідностей, то тестування вважається не пройденим.

Але виникає питання щодо кількості висловлювань, які повинна обрахувати система, щоб визначити відповідність рівня підготовки працівника. Тобто, треба знайти середнє значення \bar{i} , яке дозволить завершити процес тестування.

Для цього слід визначити безумовні ймовірності $P_i^{\delta y}$ успіху проходження тесту з висловлюванням Y_i . Деревовидна форма події, що дозволяє візуалізувати розподіл умовних ймовірностей події S_i – проходження тесту з висловлюванням Y_i , наведено на рис. 3.2.

Оскільки події незалежні, то:

$$P_i^{\delta y} = P_i \prod_{s=1}^{i-1} (1 - P_{i_s}) \quad \text{для } i = 1, \dots, n - 1,$$

$$P_n^{\delta y} = P_n \prod_{s=1}^{n-1} (1 - P_{i_s}) + \prod_{s=1}^n (1 - P_{i_s}).$$

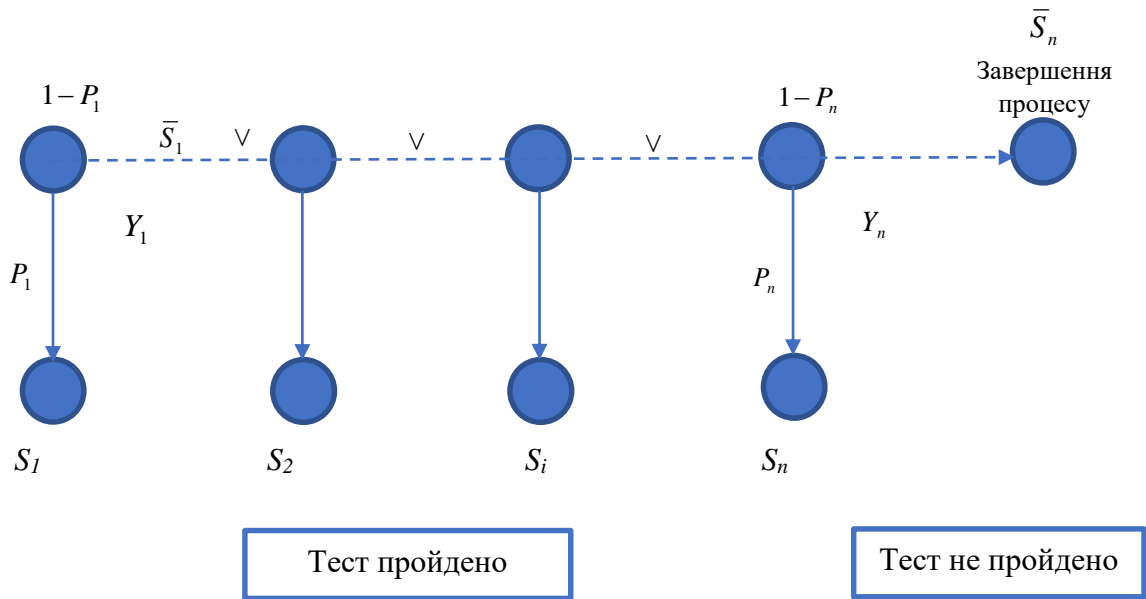


Рисунок 3.2 – Деревовидна форма події проходження тесту
з висловлюванням Y_i

Сума безумовних ймовірностей всіх подій щодо проходження тесту до завершення процесу повинна дорівнювати 1. Процес буде завершений у разі $S_1 \vee S_2 \vee \dots \vee S_n \vee \bar{S}_n = 1$.

Тобто:

$$P^{\delta y}[n] = \sum_{i=1}^n P_i \prod_{s=1}^{i-1} (1 - P_{i_s}) + \prod_{s=1}^n (1 - P_{i_s}) = 1. \quad (3.13)$$

Припускаємо, що $P^{\delta y}[n] = 1$. Якщо наведене твердження істинне, то $P^{\delta y}[n + 1] = 1$.

$$\begin{aligned}
P^{\delta y}[n+1] &= \sum_{i=1}^{n+1} P_i \prod_{s=1}^{i-1} (1 - P_{i_s}) + \prod_{s=1}^{n+1} (1 - P_{i_s}) = \\
&= \sum_{i=1}^n P_i \prod_{s=1}^{i-1} (1 - P_{i_s}) + P_{n+1} \prod_{s=1}^n (1 - P_{i_s}) + \prod_{s=1}^{n+1} (1 - P_{i_s}) = \\
&= \sum_{i=1}^n P_i \prod_{s=1}^{i-1} (1 - P_{i_s}) + P_{n+1} \prod_{s=1}^n (1 - P_{i_s}) + (1 - P_{n+1}) \prod_{s=1}^n (1 - P_{i_s}) = \\
&= \sum_{i=1}^n P_i \prod_{s=1}^{i-1} (1 - P_{i_s}) \\
&\quad + P_{n+1} \prod_{s=1}^n (1 - P_{i_s}) + \prod_{s=1}^n (1 - P_{i_s}) - P_{n+1} \prod_{s=1}^n (1 - P_{i_s}) = \\
&= P^{\delta y}[n] = 1.
\end{aligned}$$

Справедливість рівняння (3.13) для $n=3$ впливає з:

$$\begin{aligned}
P^{\delta y}[3] &= P_1 + P_2(1 - P_1) + P_3(1 - P_1)(1 - P_2) + (1 - P_1)(1 - P_2)(1 - P_3) = \\
&= P_1 + (1 - P_1)[P_2 + P_3(1 - P_2) + (1 - P_2)(1 - P_3)] = \\
&= P_1 + (1 - P_1)\{P_2 + (1 - P_2)[P_3 + 1 - P_3]\} = 1.
\end{aligned}$$

Виходячи з наведеного, вираз для визначення середнього значення \bar{i} , яке відповідає кроку завершення процесу, має представлення:

$$\bar{i} = \sum_{i=1}^n i P_i \prod_{s=1}^{i-1} (1 - P_{i_s}) + n \prod_{s=1}^n (1 - P_{i_s}). \quad (3.14)$$

Тобто, доповнивши виразом (3.14) алгоритм знаходження відповідності отриманої відповіді за словами-еталонами та ключовим словом завдання, можна отримати цикл, за яким відповідь респондента буде мати певний відсоток, за яким можна нарахувати кількість балів і представити кількісну оцінку рівня знань.

3.4.2 Модель співвідношення точності отриманої відповіді

Метою моделі співвідношення точності отриманої відповіді є характеристика обчисленого висловлювання, яке у підсумку дозволяє якісно оцінити відповідь респондента на питання.

Для цього створюємо модель з наступними компонентами: розміщуємо базовий словник за термінами та словами-еталонами в деякому регістрі, що утримує q^n комірок одиничної довжини, які розташовані на прямій з номерами $0 \div q^n - 1$. Вважаємо, що q – всі терміни і слова-еталони базового словника, а n – кількість слів у відповіді $a_1 \dots a_n$. Частина комірок N є активною. Тобто, для активних комірок слова відповіді $a_1 \dots a_n$ відповідають реальним словам базового словника.

Для побудови моделі приймаємо низку базових припущень:

- а) розподіл активних комірок – випадковий;
- б) Значення N , q , n відповідають $N \gg 1$, $q^n \gg N$.

Прийняті припущення дозволяють зробити висновок, що ймовірність r довільно обраної комірки у базовому словнику із вказаними параметрами N , q , n опинитися активною, дорівнює $\frac{N}{q^n}$. Вона прийматиме значення $r \ll 1$.

Зазначене можна представити у вигляді лінійки, де розподілені терміни та слова-еталони в порядку убуття точності за відповідними номерами, з яких довільно обираються терміни для співвідношення відповіді респондента (рис. 3.3).



Рисунок 3.3 – Візуалізація моделі активної комірки базового словника

Для формалізованого представлення моделі приймаються наступні позначення:

а) $A^{(x)} = \sum_{i=1}^n a_i^{(x)} \cdot q^{n-i}$ – координата довільної комірки з номером x ($x = 0, \dots, q^n - 1$);

б) $A_j = \sum_{i=1}^n a_{ij} \cdot q^{n-i}$ – координата активної комірки з номером j ($j = 1, \dots, N$);

в) $l^{(xy)} = A^{(y)} - A^{(x)}$ – довжина інтервалу між довільними комірками з номерами x, y ($y > x$);

г) $l_{j,j+1} = A_{j+1} - A_j$ – довжина інтервалу між двома сусідніми активними комірками.

Тоді середню довжину l інтервалу між двома сусідніми активними комірками можна представити наступним чином:

$$l = \frac{\sum_{j=1}^N (A_{j+1} - A_j)}{N-1} = \frac{A_N - A_1}{N-1}.$$

Приймається, що $A_1 = l/2$, $A_N = q^n - 1 - l/2$, тоді $l = \frac{q^n - 1 - l}{N-1} = \frac{q^n - 1}{N-1} - \frac{l}{N-1}$, звідки випливає $l = \frac{q^n - 1}{N}$. Враховуючи, що $q^n \gg 1$:

$$l = \frac{q^n}{N} = \frac{1}{r}.$$

Визначення ймовірності $P(m^{(xy)})$ того, що із довільно обраних $l^{(xy)}$ активними виявилися $m^{(xy)}$, випливає з наступного: значення $P(m^{xy})$ можна уявити як позитивне проходження тестування при довільному обиранні відповіді з випадваючого переліку можливих відповідей. Ймовірність того, що буде довільно обрана правильна відповідь дорівнює r . Величина $m^{(xy)}$ дорівнює кількості вгаданих відповідей $l^{(xy)}$.

Оскільки ймовірність довільно обрати правильно відповідь на чергове запитання не залежить від результату відповіді на попереднє питання і дорівнює r , то подібний процес можна описати за формулою Бернуллі [105]. Тобто:

$$P(m^{(xy)}) = C_{l^{(xy)}}^{m^{(xy)}} \cdot r^{m^{(xy)}} \cdot (1 - r)^{l^{(xy)} - m^{(xy)}}.$$

А враховуючи, що розподіл комірок випадковий, то $P(m^{(xy)}) = P(m^{(z)})$, де $z = y - x$. З чого випливає $l^{(xy)} = l^{(z)}$, $m^{(xy)} = m^{(z)}$.

А шукана залежність може бути представлена моделлю:

$$P(m^{(z)}) = C_{l^{(z)}}^{m^{(z)}} \cdot r^{m^{(z)}} \cdot (1 - r)^{l^{(z)} - m^{(z)}}. \quad (3.15)$$

Виходячи з (3.15), величина $P(m^{(z)})$ визначає ймовірність активних комірок в кількості $m^{(z)}$ на інтервалі $l^{(z)}$.

3.5 Алгоритм здійснення переходів між згенерованими вибірками питань при одночасному тестуванні декількох респондентів

Реалізація підбору, сегментації та навчання кадрів, особливо на великих підприємствах чи рекрутингових агентствах відбувається не поосібно, а з одночасною великою кількістю працівників, щодо яких відбувається тестування знань. Задачею інформаційної технології роботи з кадрами також є забезпечення багатопотокового ланцюга одночасного опитування персоналу.

Граф, наведений на рис. 3.2 дозволяє представити знання. Його також можна використовувати для міркувань та обчислення висловлювань. Якщо ув'язати це з рис. 3.3, то можна отримати розуміння схеми, тобто, – специфічних знань про механізм застосування вибірки питань для оцінювання

знань респондента. Виходячи з рис. 3.3 можна зазначити, що у кожного респондента при відповіді будуть використовуватися різні слова, за виключенням спеціальних термінів. Тож, активні комірочки будуть збігатися, але набори слів-еталонів будуть не ідентичними і до них відбудуватиметься звернення в порядку черговості розташування слів у відповіді, щодо яких відбувається порівняння. Тому для організації процесу здійснення переходів між згенерованими вибірками питань при одночасному тестуванні декількох респондентів пропонується використати підхід із застосуванням теорії випадкових процесів – системам масового обслуговування (СМО).

Кожна СМО складається з якогось числа обслуговуючих одиниць – каналів обслуговування. У даному випадку каналом обслуговування може вважатися робоча точка з розгорнутим інтерфейсом для проведення тестування респондента, тобто, багатоканальна СМО.

Випадковий характер проведення опитування персоналу призводить до того, що в якісь періоди часу на вході СМО скупчується надмірно велике число заявок. Процес роботи у зазначеному випадку стає випадковим процесом з дискретними станами і безперервним часом. Стан системи буде змінюватися стрибкоподібно в моменти появи якихось подій:

- а) підключення нового респонденту до системи;
- б) вихід респондента із системи;
- в) затримка при пошуку відповідностей у системі.

У зазначеному випадку система тестування знань працівників представлятиме собою n -канальну СМО з необмеженою чергою, в якій нумерація станів буде формуватися по числу заявок, що знаходяться в системі:

- а) S_0 – в СМО заявок немає (усі канали вільні);
- б) S_1 – зайнятий один канал, інші вільні;
- в) S_2 – зайнято два канали, інші вільні;

.....

S_k – зайнято k каналів, інші вільні;

.....

S_n – зайняті усі n каналів (черги немає);

S_{n+1} – зайняті усі n каналів, один з респондентів стоїть в черзі (що є небажаним результатом для системи оцінювання знань кадрів);

S_{n+r} – зайняті усі n каналів, r респондентів стоять в черзі (система не справилася з навантаженням).

Граф станів такої системи наведено на рис. 3.4.

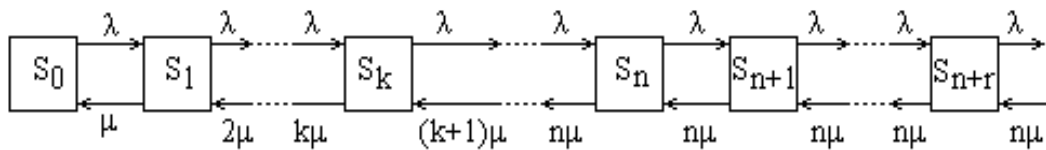


Рисунок 3.4 – Граф станів n -канальної СМО

На рис. 3.4 використані наступні позначення:

а) перехід із стану в стан за рахунок вступу заявок з інтенсивністю λ (верхні стрілки);

б) перехід із стану в стан за рахунок обслуговування заяво :

– перехід із стану S_1 в S_0 (звільняється єдиний зайнятий канал) здійснюється з інтенсивністю λ ;

– перехід із стану S_2 в S_1 (у системі знаходилися дві заявки, виконується одна) двома способами: йде або перша, або друга заявка. Іntenсивність переходу 2μ ;

– перехід із стану S_k в S_{k-1} $(k+1)\mu$ можна виконати k способами: обслуговується або перша, або друга, ..., або k -та заявка;

в) перехід зі стану S_{n+1} в S_{n+r} $(r>1)$ можна виконати тільки n способами (оскільки усі канали залишаються зайнятими). Іntenсивність переходу $n\mu$;

Наведений граф (рис. 3.4) є схемою загибелі і розмноження з нескінченним числом станів. Природна умова існування фінальної ймовірності описується нерівністю:

$$\frac{\rho}{n} < 1 \quad (3.16)$$

Якщо $\rho/n > 1$ – черга росте до нескінченності.

Якщо умова (3.16) виконана, то фінальна ймовірність існує. В цілому, застосування зазначеного підходу можливо проілюструвати задачею (Додаток В). Аналізуючи результати моделювання, був отриманий висновок щодо деякої парадоксальності результатів. Це пояснюється тим, що в першому варіанті двоканальної СМО менше середня доля часу, яку простоєє кожна з робочих точок (комп'ютерів, на якому відбувається тестування): якщо респондент А замислився над питанням, а респондент Б вже відповідає на друге питання, то система переходить до перевірки результату респондента Б і навпаки. Другий варіант не передбачає такої заміни – йде очікування результату від респондента А, потім від респондента Б. До того ж в розглянутому прикладі СМО працюють на межі своїх можливостей: варто трохи збільшити час очікування результатів відповіді (тобто зменшити μ), як вони вже перестануть справлятися з потоком респондентів, і черга почне необмежено зростати. А «зайві простої» системи в якомусь сенсі рівносильні зменшенню її ефективності μ .

Таким чином, що здається спочатку парадоксальним (або навіть просто невірним), результат обчислень виявляє правильним і пояснюючим.

Проте наведений метод заснований на схемі загибелі-розмноження і формулі Літтла і не дозволяє розглянути випадковий процес, що виник в результаті використання засобів штучного інтелекту і його еволюція залежить від стану до настання деякого часу t . Зазначене розглядається через немарківські процеси, для яких існують тільки окремі, прочитані результати, що дозволяють виразити в явному, аналітичному виді характеристики СМО через задані умови завдання – число каналів, характер потоку, вид, розподіл часу обслуговування.

Вище була розглянута n -канальна СМО з відмовами, з простим потоком заявок і довільним розподілом часу обслуговування. Якщо використати формулу Ерланга [106], то можна отримати фінальні ймовірності для станів СМО з відмовами і можна здійснити перехід до формули Поллачека-Хічкина [108], що дозволить організувати процес з мінімізацією затримки у часі. Проте у разі багатоканальних немарківських СМО точних аналітичних методів не існує. Але реалізація сучасних систем підбору, сегментації та навчання кадрів з використанням засобів ШІ, наприклад, на веб-платформі, передбачає систему роботи саме подібну багатоканальній немарківській СМО. Тож основним методом розробки таких систем є статистичне моделювання.

Загальним для усіх алгоритмів комп'ютерного моделювання СМО є організація «плину часу» у вигляді окремих відрізків часу, наприклад, секунд (якщо середній час перебування стану вимірюється в хвилинах) або умовних тактів. На тлі цього часу, тобто, кількості минулих відрізків, організовується імітація подій, що відбуваються в системі.

Для реалізації поставленої задачі пропонується провести моделювання СМО на прикладі наступного завдання: є три робочі точки ($n=3$), інтенсивність потоку обробки результатів тестування 1,5 хв., середній час перевірки результатів одного тестування: $\bar{t}_{об} = 2$ хв. Усі потоки приймаємо простими. Необхідно знайти характеристики ефективності такої СМО: середню відносну пропускну спроможність і середнє число зайнятих каналів.

Завдання можна вирішити за наступним алгоритмом:

1) Початок алгоритму: ввід початкових даних:
 – число випадкових випробувань NW ;
 – розрахунок приведеної інтенсивності (для проведення розрахунків важливі не самі значення інтенсивностей, а їх співвідношення, яке дорівнюватиме за даною задачею 3; відповідно, відношення середнього часу обслуговування до середнього часу вступу заявок також дорівнюватиме 3).

2) Визначається масив $canal(1...3)$, в елементах якого фіксуватиметься час виконання обробки результатів відповідного каналу.

3) На початковому (нульовому) такті:

- генерується період часу (у тактах) між першою і другою заявкою, як випадкове число, розподілене за показовим законом;
- отримане значення розміщується комірку змінної часу $Time_Z$;
- генерується період часу (у тактах) обробки отриманих результатів, які розміщуються в першому елементі масиву $canal$;
- число випробувань NW зменшується на 1;
- визначаються і обнуляються наступні змінні:
 - а) n_s – лічильник числа зайнятих каналів;
 - б) N_otw – лічильник числа відхилених результатів;
 - в) N_ser – лічильник числа перевірених і прийнятих результатів;
 - г) k – лічильник тактів часу;
- проводяться наступні обчислення до отримання $NW = 0$:
 - а) збільшуємо k : $k=k+1$ («хід часу»)
 - б) перевірка вмісту лічильника часу $Time_Z$. Якщо $Time_Z = 0$:
 - генерується новий відрізок часу до надходження нових результатів для обробки;
 - отримане значення розміщується в комірці $Time_Z = 0$;
 - пошук вільного каналу, для якого $canal(i)=0$. Якщо вільних каналів немає – збільшуємо на 1 лічильник N_otw ;
 - якщо є вільний канал, то генерується випадковий час обслуговування, додається до нього 1 і отриманий результат розміщується у виявленій вільній елемент масиву $canal$. Одночасно збільшується на 1 вміст лічильника N_ser ;
 - зменшується на 1 вміст лічильника NW і відбувається перехід до пункту г);
 - в) якщо при аналізі за пунктом б) $Time_Z > 0$ – зменшується вміст цього лічильника на 1 («хід часу»);
 - г) відбувається перевірка значення каналів. Якщо $canal(i)>0$ – зменшується вміст каналу на 1 і, одночасно, збільшується на 1 вміст

лічильника зайнятих каналів n_s . Якщо $\text{canal}(i)=0$ – вміст каналу не змінюється.

4) Після закінчення обчислень розраховуються результати:

- пропускної спроможності системи (N_{ser}/NW);
- середнього числа зайнятих каналів (n_s/k).

Нижче приведений варіант лістингу програми на мові VBA для Excel (на лістингу приведені налагоджувальні оператори, відокремлені апострофом):

```

Sub Erlang()
Dim canal(1 To 3) As Integer
Randomize
NW = Cells(4, 2)
n_w = NW
time_z = Int(- 5 * Log(1 - Rnd))
' MsgBox time_z
n_otw = 0
n_ser = 1
k = 0
n_s = 0
canal(1) = Int(- 15 * Log(1 - Rnd))
NW = NW - 1

While NW > 0
k = k + 1
time_z = time_z - 1
If time_z <= 0 Then
'          'Stop
time_z = Int(- 5 * Log(1 - Rnd))

i = 0

NW = NW - 1

```

```
lab2:
i = 4

If canal(1) = 0 Then
i = 1
GoTo lab3
End If
If canal(2) = 0 Then
i = 2
GoTo lab3
End If
If canal(3) = 0 Then
i = 3
GoTo lab3
End If
lab3:
If i <= 3 Then
canal(i) = Int(- 15 * Log(1 - Rnd)) + 1
n_ser = n_ser + 1

Else
n_otw = n_otw + 1
End If
End If
For i = 1 To 3

If canal(i) > 0 Then
canal(i) = canal(i) - 1
n_s = n_s + 1
End If
Next i

'   'Збереження вмісту каналів у кожний такт
'   ' час для відладки
'   'Cells(10 + k, 1) = k
'   'Cells(10 + k, 2) = time_z
'   'For i = 1 To 3
```

```

'      'Cells(10 + k, 2 + i) = canal(i)
'      'Next i

Wend

Cells(4, 1) = k
Cells(4, 2) = n_w
Cells(4, 3) = n_otw
Cells(4, 4) = n_ser
For i = 1 To 3
Cells(8, i) = canal(i)
Next i
Cells(7, 5) = n_s / k
End Sub

```

Результати перевірки наведеного алгоритму та моделювання відносної пропускної спроможності (A) і середнього числа зайнятих каналів ($k_{сер}$) при значеннях $NW = 100, 1000, 10000$ приведені в таблиці 3.4.

Таблиця 3.4 – Результати перевірки наведеного алгоритму

| | $NW=100$ | | $NW = 1000$ | | $NW = 10000$ | |
|------------------|----------|-----------|-------------|-----------|--------------|-----------|
| | A | $k_{сер}$ | A | $k_{сер}$ | A | $k_{сер}$ |
| 1 | 0,630 | 1,910 | 0,629 | 1,960 | 0,638 | 2,060 |
| 2 | 0,600 | 2,100 | 0,660 | 2,000 | 0,626 | 2,080 |
| 3 | 0,650 | 2,130 | 0,651 | 2,060 | 0,632 | 2,060 |
| 4 | 0,710 | 1,950 | 0,653 | 2,040 | 0,639 | 2,080 |
| 5 | 0,510 | 2,190 | 0,634 | 2,050 | 0,639 | 2,060 |
| Середнє значення | 0,620 | 2,056 | 0,645 | 2,022 | 0,635 | 2,068 |

Як впливає з табл. 3.4, в усіх випадках виходять відносно постійні значення, що вказує на адекватність результатів моделювання і аналітичного розрахунку.

Якщо ж прийняти, що час розподілений згідно із законом, відмінним від показового, наприклад, рівномірного розподілу, то можна внести зміни до операторів програми, які розраховують випадковий час (генерується за

формулою: $Time_Z = \text{int}(5 * \text{rnd} + 1)$). Результати моделювання наведені в табл. 3.5.

Таблиця 3.5 – Результати моделювання за умов рівномірного розподілу часу на обробку результатів тестування всіх респондентів системи

| | NW=100 | | NW = 1000 | | NW = 10000 | |
|------------------|--------|-------|-----------|-------|------------|-------|
| | A | kсер | A | kсер | A | kсер |
| 1 | 0,650 | 2,490 | 0,615 | 2,526 | 0,615 | 2,521 |
| 2 | 0,610 | 2,616 | 0,621 | 2,512 | 0,617 | 2,521 |
| 3 | 0,680 | 2,440 | 0,631 | 2,500 | 0,615 | 2,511 |
| 4 | 0,660 | 2,462 | 0,605 | 2,515 | 0,614 | 2,517 |
| 5 | 0,650 | 2,398 | 0,602 | 2,508 | 0,617 | 2,524 |
| Середнє значення | 0,650 | 2,481 | 0,615 | 2,512 | 0,616 | 2,519 |

З порівняння табл. 3.4 та 3.5 для рівномірного розподілу пропускна спроможність дещо менша, а середнє число зайнятих каналів дещо більше, ніж для показового розподілу.

Варто зазначити, що для моделювання можна використовувати і інші, ще складніші розподіли ймовірності, у тому числі – емпіричні розподіли, отримані з експериментальних даних, тобто, в тих випадках, коли аналітичне рішення свідомо неможливе.

3.6 Висновки за розділом 3

В результаті застосування логіко-ймовірнісного підходу до формування множини завдань для незалежних випробувань в системах відбору персоналу були отримані наступні результати:

1) в роботі було розвинуто підхід Г.П. Донця, В.А. Пепеляєва, О.М. Трофимчука про оптимальне розбиття на кількість груп та запропоновано стратегію щодо генерування вибору випадкового питання для формування множини питань в системі тестування знань працівника. Створено тестовий

примірник програми-генератора та апробовано при формуванні вибірки з десяти випадкових завдань за номерами запису від 1 до 2000;

2) розроблено алгоритм за оберненою функцією та за методом Неймана-Пірсона для генерації випадкового номера завдання за заданої аналітичної функції розподілу;

3) запропоновано для створення системи підбору, сегментації та навчання кадрів з використанням засобів штучного інтелекту використати логічний підхід із застосування синтаксису обчислювання висловлювань для вирішення задач відповідності отриманої відповіді поставленому питанню. Наведено сукупність правил логіки, а також описано процес визначення відповідності відповіді поставленому питанню за допомогою дерева висловлювань. Наведено алгоритм знаходження відповідності отриманої відповіді за словами-еталонами та ключовим словом завдання;

4) наведена модель процесу обчислення висловлювань з використанням базового словника, який сформовано за правилами логіки, що дозволяє з використанням засобів штучного інтелекту визначити, хибна чи істинна була відповідь. Для цього використано підхід, який у разі отримання неточної відповіді та невідповідності з висловлюванням Y_i , здійснює перехід до нового сформованого висловлювання Y_{i+1} , яке також не є абсолютно точним, але дозволяє співставити відповідь респондента з отриманням оцінки знань. При вірній відповіді та співпадіння з висловлюванням Y_{i+1} , процес завершується.

Представлена модель співвідношення точності отриманої відповіді для якісної оцінки знань;

5) наведено алгоритм здійснення переходів між згенерованими вибірками питань при одночасному тестуванні декількох респондентів за підходом, використовуваному у СМО. Враховано, що тестування відбувається найчастіше для груп осіб, а не поодиноких респондентів, тому є необхідність забезпечення багатопотокового ланцюга одночасного опитування персоналу. Зроблено перехід від моделі процесу обчислення висловлювань та якісної оцінки знань до забезпечення системою процесу такого перебору питань, який

дозволяє формувати різні вибірки для кожного окремого респондента. Запропоновано для організації процесу здійснення переходів між згенерованими вибірками питань при одночасному тестуванні декількох респондентів використовувати підхід із застосуванням теорії систем масового обслуговування. Представлено алгоритм та лістинг програми.

Особливістю вирішених задач є те, що запропонований підхід може бути використаний в розробці систем управління персоналом із використанням засобів штучного інтелекту як через прикладні додатки на персональних комп'ютерах, що було наведено в роботі, так і реалізованих на веб-платформах, що буде представлено у наступному розділі роботи;

4) матеріали, що подані в розділі, знайшли часткове відображення в роботах [108, 109].

РОЗДІЛ 4

РОЗРОБКА ІНСТРУМЕНТАЛЬНОГО ЗАСОБУ ДЛЯ ПРИЙНЯТТЯ РІШЕНЬ ЗА УМОВ НЕВИЗНАЧЕНОСТІ ПРИ ВІДБОРІ КАДРІВ ЧЕРЕЗ ІНТЕРНЕТ-ПЛАТФОРМУ

4.1 Реалізація механізму відтворення людської поведінки при підборі кадрів за умов невизначеності

Для того, щоб за допомогою системи тестування можна було здійснити вибір питання, був проведений експеримент з комп'ютерного моделювання. Сутність експерименту полягала у наступному.

За допомогою апаратної обчислювальної платформи для конструювання Arduino [110], експеримент був реалізований в онлайн-програмі для 3D-моделювання Tinkercad [111], де був створений пристрій, в основі якого покладено автомат Мура. Цей пристрій (рис. 4.1) імітував поведінку старшої людини при спілкуванні з дитиною 3-4 років, для з'ясування того, чи запам'ятала дитина певний, необхідний для подальшого розвитку факт. Логічна схема такої взаємодії полягає у наступному:

- а) доросла людина задає питання;
- б) якщо відповідь не вірна – «хиба», повтор питання, щоб дитина подумала і відповіла знову;
- в) якщо відповідь вірна – «істина», і відбувається перехід до нового питання.

Аналогічний підхід реалізовується і при тестуванні, у випадку, коли дається можливість двічі або тричі відкоригувати свою відповідь (тобто, здійснити цикл, що дозволяє виправити неправильну відповідь).

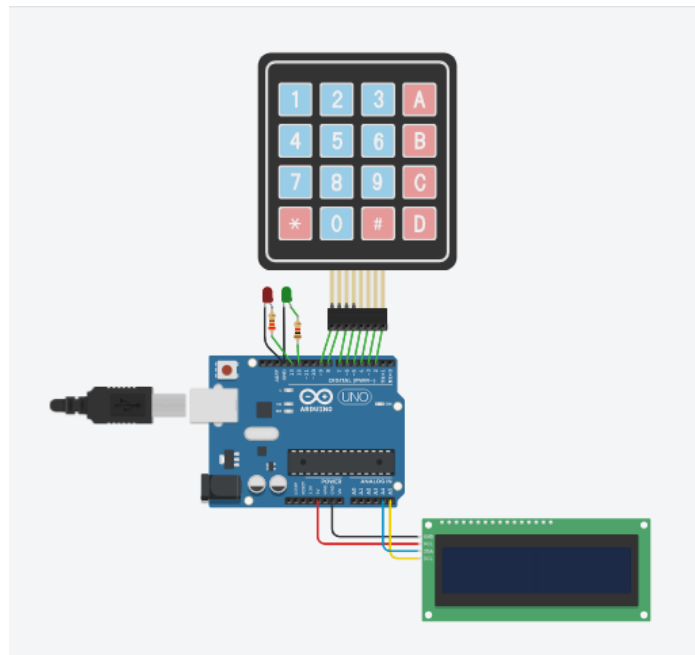


Рисунок 4.1 – Експеримент з комп’ютерного моделювання: створення пристрою, який імітує поведінку дорослої людини при навчанні дитини

В Додатку Г наведена реалізація комп’ютерного експерименту з використання абстрактного автомату для створення системи імітації людської поведінки в процесі навчання, зокрема: наведена схема, за якою створено пристрій (Додаток Г. 1), перелічені використані компоненти (Додаток Г. 2), та наведений лістинг коду (Додаток Г. 3).

У підсумку, експеримент дозволив опрацювати механізм переходів між питаннями в залежності від правильної відповіді на попереднє питання. На цій основі було розроблене 3D резюме для тестування знань працівника (за приклад взяті вимоги до програміста C++).

Початком роботи із 3D резюме є реєстрація в системі. За зареєстрованим логіном фіксується унікальна ознака об’єкта (ID). За цим ID записи про дії користувача вносяться до бази даних. Кожен вхід до системи через спеціальну форму входу (рис. 4.2) фіксується за датою, часом входу і виходу, виконаними або не вірно виконаними завданнями.

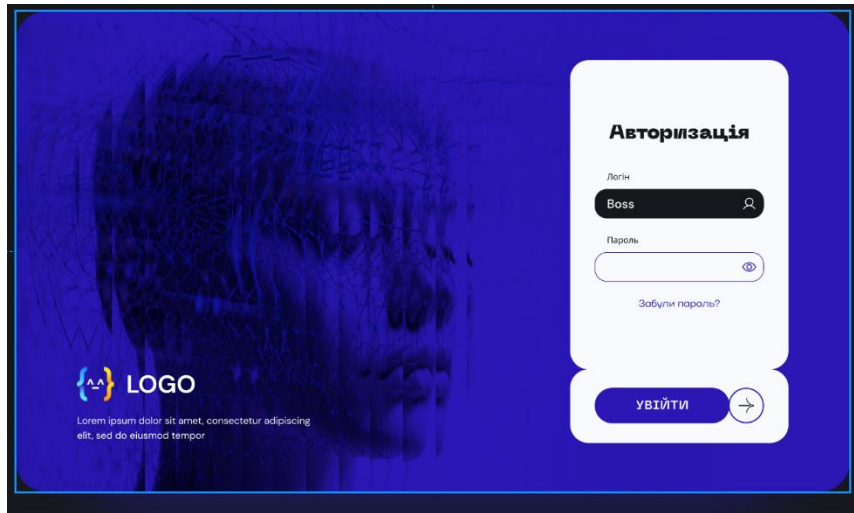


Рисунок 4.2 – Авторизація в системі 3D резюме

Точка входу за часом в систему 3D резюме дозволяє відкрити початковий етап тестування – тестове письмове завдання. Такий підхід дозволяє за ключовими словами на самому початку складання 3D резюме визначити, наскільки людина володіє тезаурусом спеціальності та може в подальшому розібратися з іншими завданнями, необхідними для визначення компетентностей працівника.

Такі ключові слова утримуються в базовому словнику системи 3D резюме (див. підр. 3.3.2). Сутність реалізації відбору на цьому етапі тестування полягає у наступному. Приймаємо, що на основі (3.13) – (3.15) вірно:

$$C_{l^{(z)}}^{m^{(z)}} \cdot r^{m^{(z)}} \cdot (1 - r)^{l^{(z)} - m^{(z)}} = P(m^{(z)}, r, l^{(z)}). \quad (4.1)$$

Нехай при цьому у (4.1) виконується $l^{(z)} = q^{n-i}$. Позачення i виступає кількістю старших розрядів значення координати $A^{(z)} = l^{(z)}$. А враховуючи рис. 3.3, $m^{(z)} = g_i$ є випадковою кількістю активних комірок у обраному інтервалі. Для визначення співвідношення між i та g_i абстрактний процес написання саме потрібного ключового слова при виконанні завдання можна ускладнити наступним чином.

Припустимо, що при вирішенні завдання користувач 3D резюме пише послідовно декілька q^{n-i} ключових слів ($i = 1, 2, \dots, n$). Якщо на кроці $i = v$ виконане завдання відправляється на перевірку у систему і виявляється максимально можлива кількість використаних ключових термінів $g_i = \dot{m}$, процес тестування за даним завданням можна вважати закінченим. Проте процес виконання завдань повторюється багаторазово. Для певного висновку щодо відповідності користувача висунутим до вакантної посади вимогам слід визначити середнє значення $v_{cp}(g_i = \dot{m})$. Це значення буде ознакою завершення процесу при багаторазових випробуваннях.

Тобто, перше тестування у 3D резюме передбачає вказівку одного з ключових слів, що внесені до базового словника системи стосовно конкретної вакантної посади. І успіхом виконання завдання є вказівка одного чи декількох ключових слів, а умови Y_i – у завданні значення q^{n-i} .

Тоді, враховуючи зазначене та вираз (3.14) отримуємо механізм умови здійснення відбору фахівця:

$$v_{cp}(g_i = \dot{m}) = \sum_{i=1}^n i \cdot P_i^{\delta y} =$$

$$= \sum_{i=1}^n i \cdot P(g_i = \dot{m}) \prod_{s=1}^{i-1} [1 - P(g_s = \dot{m})] + n \prod_{s=1}^n [1 - P(g_s = \dot{m})],$$

де:

$P(g_{i,s} = \dot{m})$ – умовна ймовірність того, що на кроці i,s в завданні зазначається \dot{m} ключових слів. Для $\dot{m} > q^{n-i}$ величина $P(g_{i,s} = \dot{m}) = 1$. А якщо врахувати (3.15) значення умовних ймовірностей $P(g_i = \dot{m})$ та спрощуючи позначення відповідно з (4.1), можна отримати:

$$v_{cp}(g_i = \dot{m}) = \sum_{i=1}^n i \cdot P(\dot{m}, r, q^{n-i}) \cdot \prod_{s=1}^{i-1} [1 - P(\dot{m}, r, q^{n-s})] + \quad (4.2)$$

$$n \prod_{s=1}^n [1 - P(\dot{m}, r, q^{n-s})],$$

де

$$P(\dot{m}, r, q^{n-i}) = C_{q^{n-i}}^{\dot{m}} \cdot r^{\dot{m}} \cdot (1 - r)^{q^{n-i} - \dot{m}}. \quad (4.3)$$

Вираз (4.2) визначає середнє значення i , при якому в інтервалах $0 \div q^{n-1}, 0 \div q^{n-2}, \dots, 0 \div q^0$ виявиться саме \dot{m} активних комірок, що утримують ті завдання, які відповідають заданим ключовим словам.

Наприклад, користувачу відкривається звичайна задача, яку слід реалізувати на мові C++ стосовно типів даних, змінних та констант» «Об'явити змінні, за допомогою яких можна буде порахувати вартість покупки». Стандартна відповідь міститиме приблизно такий фрагмент коду:

```
#include
#include
#include
using namespace std;
int main()
{
SetConsoleCP(1251);
SetConsoleOutputCP(1251);
int товар 1, товар 2, товар 3, товар 4, товар 5;
cout << "слід ввести кількість" <> товар 1;
cout <<"кількість товар 1 "<< товар 1 << endl;
...
cout << "-----" << endl;
float Pricетовар 1, ..., .....;
cout << "ціна товар 1 " <> Pricетовар 1;
...
float sum = 0;
sum = формула;
```

```
cout << "Загальна вартість = ";
cout << реалізація формули << sum << endl;
}
```

Звичайно що, у наведеному прикладі ключовими словами виступлять терміни стандартної бібліотеки мови C++. Якщо користувач 3D резюме їх застосував при відповіді, то це вказує на наявні знання, необхідні для отримання вакантної посади. Цими словами можуть виступати float, sum, cout, include.

На рис. 4.3 наведений скріншот реалізації такого підходу у 3D резюме, коли активною є лише вкладка з тестовими завданнями, які дозволять на першому етапі виявити відповідність кандидата на посаду, що є вакантною.

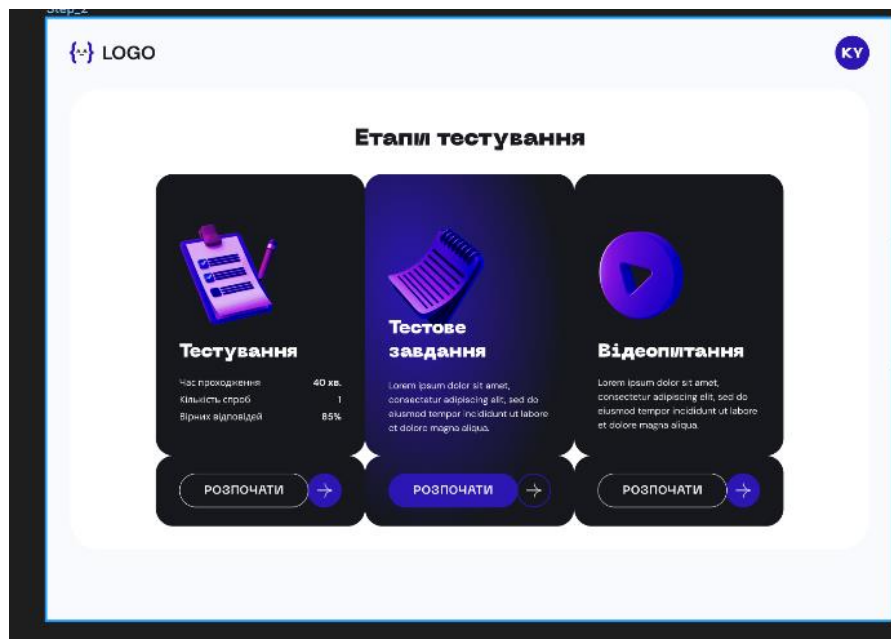


Рисунок 4.3 – Вкладки 3D резюме з активною кнопкою доступу до відбору через тестові завдання

Безпосередньо інтерфейс роботи над тестовим завданням є класичним вікном для введення відповіді користувачем своєю рідною мовою чи, на необхідності, для написання програмного коду (рис. 4.4).

Після того, як написання тестового завдання виконано, активуються кнопки двох інших вкладок 3D резюме – проходження тестів та відеопитання, для проведення опитування у відеорежимі.

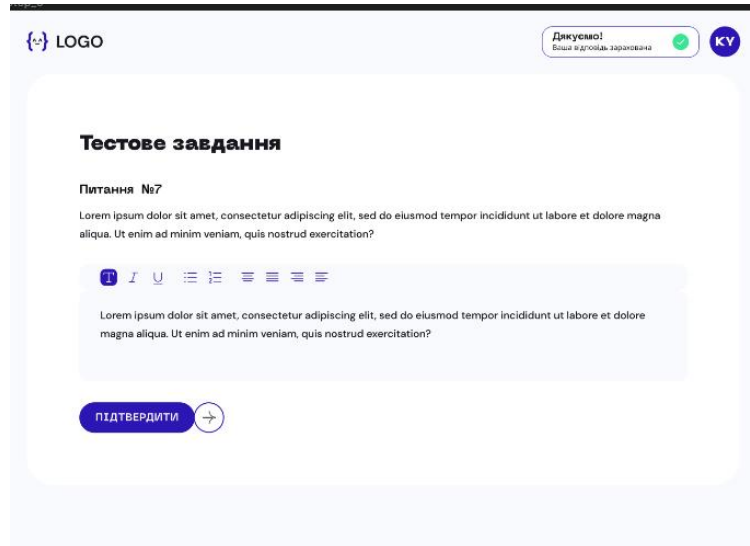


Рисунок 4.4 – Інтерфейс користувача для роботи над тестовим завданням

Приклад інтерфейсу для проведення тестування наведено на рис. 4.5. Вибір тестів для тестування відбувається за результатами виконання тестового завдання – тести вибираються таким чином, щоб використані раніше користувачем ключові слова не повторювалися у тестах.

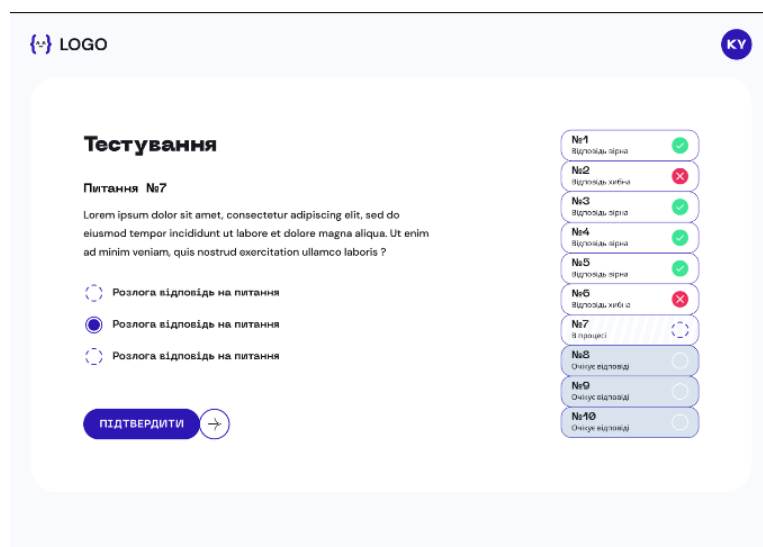


Рисунок 4.5 – Інтерфейс проходження тестів у 3D резюме

Запис результатів проходження завдань відбувається у базу даних 3D резюме за кожним завданням з позначенням часу виконання. Збільшення або зменшення часу виконання при порівнянні результатів проходження тестів також дає можливість проаналізувати, наскільки людина покращила свої знання та навички у порівнянні з попереднім тестуванням.

Відеопитання, реалізація інтерфейсу у 3D резюме якого наведена на рис. 4.6, дозволяє здійснити відповідь на поставлене питання так, ніби йде розмова з менеджером по персоналу в режимі реального часу через засоби комунікації. Однак реалізація цього інструменту відбувається через відеозапис питання та відеозапис відповіді для подальшого аналізу за тими ж ключовими словами.

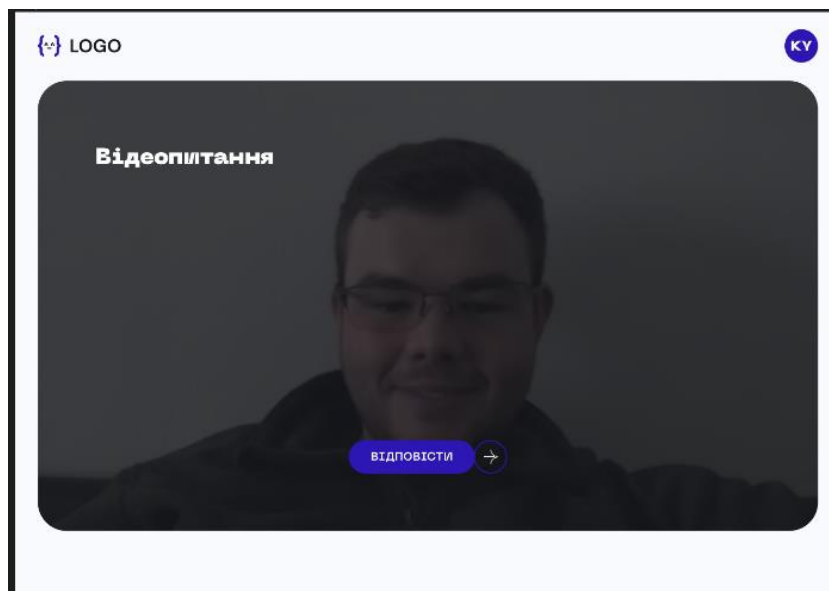


Рисунок 4.6 – Реалізація інтерфейсу відеопитання у 3D резюме

Особливістю цього інструментарію у 3D резюме полягає в тому, що саме цей фрагмент подається на розгляд менеджеру з персоналу у звіті за кожним працівником, що проходив опитування. За іншими питаннями подаються лише результати і при необхідності – порівняння чи набрані бали. Відеопитання дозволяє оцінити менеджеру з персоналу те, чого не розпізнає система – емоційний настрій, впевненість у собі, вміння триматися при виконанні завдання.

Лістинг коду реалізації 3D резюме наведено в додатку Д посторінково (Д 1 – Д 4).

4.2 Реалізація процедур заміни номеру питання та тригерів у базі даних 3D резюме

В процесі роботи кожного користувача, як зазначалося вище, за індивідуальним ID відбувається запис в базі даних про результати проходження 3D резюме. У підсумку, після першого тестування, з'являються записи щодо виконаних та невиконаних завдань (див. рис. 4.5). Наступна фіксація цього ж самого користувача за ID в системі 3D резюме для перевірки знань та навичок, повинна викликати створення вибірки питань за виключенням тих, на які були дані правильні відповіді. Це питання можна вирішити за допомогою створення копій таблиць у системі керування реляційними базами даних MySQL, записи в яких буде змінено, та задаванням нових значень (номерів питань) для зміни вмісту.

Наприклад, копіюються дві таблиці – member (copy_member), де записані дані про зареєстрованих користувачів 3D резюме та adult (copy_adult), що утримують дані про виконані завдання:

```
use library;
select * into copy_member from member.
```

Просте клонування можна здійснити також за допомогою команди:

```
CREATE TABLE new_table SELECT * FROM original_table.
```

Після цього створюється новий робочий процедури заміни вмісту таблиці бази даних. Наприклад, здійснюємо заміну одного питання іншим за номером у базі даних завдань:

```
USE [Respondents]
GO
```



```

/***** Object:  StoredProcedure
[dbo].[ChangeQuestionNumber]      Script Date: *****/
SET ANSI_NULLS ON
GO
SET QUOTED_IDENTIFIER ON
GO
-- =====
-- Author:
-- Create date:
-- Description:  Процедура, яка дозволяє змінити Question
Number
-- =====
ALTER PROCEDURE [dbo].[ChangeQuestionNumber]
    -- Add the parameters for the stored procedure here
    @member int = 0,
    @number varchar(250) = ''
AS
BEGIN
    -- SET NOCOUNT ON added to prevent extra result sets
from
    -- interfering with SELECT statements.
    SET NOCOUNT ON;

    -- Insert statements for procedure here
    SELECT *
    from copy_member
    update copy_member
    set QuestionNumber = @number
    where member_no = @member
END

```

При виконанні наведеного коду отримується:

```

use library;
exec QuestionNumber @member = 13, @number = '21'.

```

При створенні бази даних є важливим розробка триггеру, що дозволяє здійснити варіанти переходів між окремими питаннями в залежності від того, на яке питання була дана правильна відповідь, а на яке – ні. Знов таки, відбувається робота у робочих (дубльованих таблицях), бо вони не повинні мати реляційних зв'язків (для забезпечення коректної роботи бази даних). Запит вибірки в таблицю (select * into from ...) виконує лише вибірку даних без копіювання будь-яких інших обмежень цілісності сутностей чи цілісності посилань.

Наступним кроком створюємо тригер для каскадного витирання рядків підлеглої таблиці, де зазначені завдання для проходження тесту:

```

USE [.....]
GO
/***** Object:  Trigger [dbo].[CascadeWiping]      Script
Date: 14.11.2022 0:00:42 *****/
SET ANSI_NULLS ON
GO
SET QUOTED_IDENTIFIER ON
GO
-- =====
-- Author:
-- Create date:
-- Description:      Тригер для каскадного витирання рядків
підлеглої таблиці
-- =====
ALTER TRIGGER [dbo].[CascadeWiping]
    ON [dbo].[copy_member]
    AFTER DELETE
AS
BEGIN
    -- SET NOCOUNT ON added to prevent extra result sets
from
    -- interfering with SELECT statements.
    SET NOCOUNT ON;

```

```

-- Insert statements for trigger here
delete from copy_adult
from copy_adult as p inner join deleted as d
on p.member_no = d.member_no;
END

```

Аналогічно будується алгоритм для вставлення нового рядка, що містить інформацію про нове обране системою питання.

Також для роботи є важливим створення тригера, який реалізуватиме правило бізнес-логіки для бази даних:

```

-- =====
-- Template generated from Template Explorer using:
-- Create Trigger (New Menu).SQL
--
-- Use the Specify Values for Template Parameters
-- command (Ctrl-Shift-M) to fill in the parameter
-- values below.
--
-- See additional Create Trigger templates for more
-- examples of different Trigger statements.
--
-- This block of comments will not be included in
-- the definition of the function.
-- =====
SET ANSI_NULLS ON
GO
SET QUOTED_IDENTIFIER ON
GO
-- =====
-- Author:
-- Create date:

```

```

-- Description:      <тригер, який реалізуватиме правило
бізнес-логіки для бази даних>
-- =====
CREATE TRIGGER BusinessLogic
    ON copy_member
    AFTER INSERT
AS
BEGIN
    -- SET NOCOUNT ON added to prevent extra result sets
from
    -- interfering with SELECT statements.
    SET NOCOUNT ON;

    -- Insert statements for trigger here
    declare @member_no int
    select @member_no = member_no from copy_member
    if (@member_no <= 0)
    begin
        RAISERROR ('The reader code is entered incorrectly.
The code cannot be negative', 10, 1)
        ROLLBACK TRANSACTION
    end
END
GO

```

Наведений тригер не дозволить вносити дані з неправильними значеннями при заповненні таблиць бази даних, формуванні нових або зміні існуючих, додавання до бази даних інформації, не пов'язаної з тематикою роботи 3D резюме.

4.3 Тестування експериментальної версії 3D резюме

4.3.1 Імітаційна модель інтерфейсу 3D резюме з тестуванням користувача за ключовим словом

Розроблена модель 3D резюме за своєю суттю є імітаційною або експериментальною моделлю 3D резюме, оскільки утримує в собі базові елементи з окремими засобами ШІ – можливістю здійснювати вибір в залежності від відповіді, збереження результатів, порівняння для використання результатів у наступному виборі. Базовий словник 3D резюме має обмежену кількість ключових слів. Перелік питань складається із загальнонаукових, за складністю призначених для випускника середньої школи.

Тестування робочої версії 3D резюме відбувалося шляхом опитування випадкових респондентів у складі 100 осіб. Їх просили відповісти на питання не обмежуючи час виконання завдання. Обмеженням тестування виступав перебір питань за принципом механізму імітації поведінки (див. рис. 4.1) – тобто, нове питання з'являлося тоді, коли була отримана правильна відповідь на попереднє запитання.

Крім того, проводилися розрахунки оцінки інтерфейсу 3D резюме. При цьому враховувалося, що результати експерименту та результати розрахунків будуть мати відхилення. Це обумовлено наближеністю експериментальних оцінок інтенсивності праці окремих операції при роботі з персональним комп'ютером, що складали сумарну інтенсивність праці H , та прийняті в експерименті спрощення.

Варто зазначити, що вплив сумарних складових інтенсивності праці на результати оцінок, взагалі є суттєвим для технології GOMS. Для абсолютних оцінок зазначений вплив можна мінімізувати за допомогою чіткого розмежування кваліфікації користувача комп'ютера завданням, що подаються

у 3D резюме. Для оцінки варіанту інтерфейсу цей вплив можна не враховувати.

Мета імітаційного моделювання для оцінки інтерфейсу розробленої версії 3D резюме полягала у наступному:

а) перевірка адекватності сприйняття інтерфейсу 3D резюме різними за рівнем підготовки користувачів, уточнення оцінок інтенсивності праці окремих операцій при роботі з 3D резюме, а також апробація моделей, які вже закладені у веб-реалізацію 3D резюме;

б) оцінка часу реакції системи, коли вводиться ключове слово і коли воно не вводиться користувачем, аналіз здійснення переходів між питаннями.

У якості показників, що характеризують зазначені процеси, обрані:

а) середнє значення \bar{v}_1 кількості символів, що вводяться користувачем системи 3D резюме для визначення ключового слова;

б) середнє значення $\bar{m}_1^{(1)}$ сумарної кількості слів, які вводяться користувачем для відповіді на завдання з врахуванням можливості внесення виправлення до відповіді.

Схема взаємодії основних перелічених об'єктів у 3D резюме наведена на рис. 4.7.

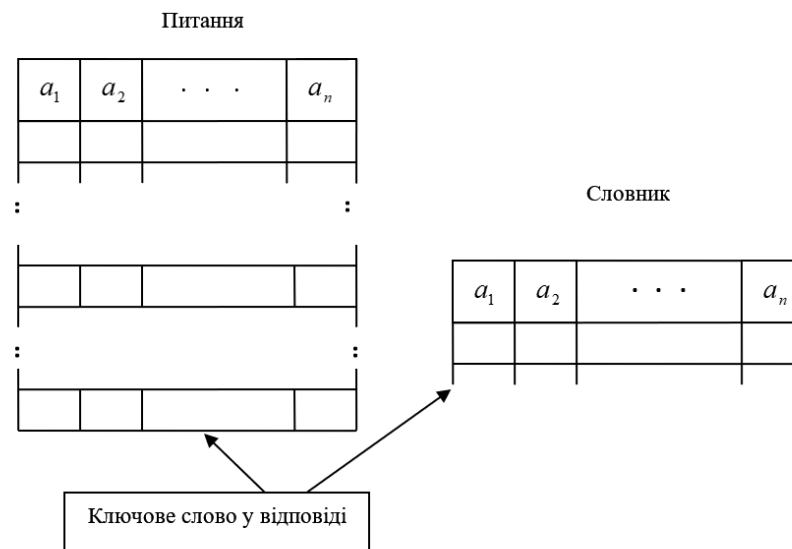


Рисунок 4.7 – Схема взаємодії між питаннями, словником базових слів та ключовим словом у відповіді у 3D резюме

В динаміці процес співвіднесення питання, словником базових слів та ключовим словом у відповіді у 3D резюме в процесі проведення експерименту наведено на рис. 4.8.

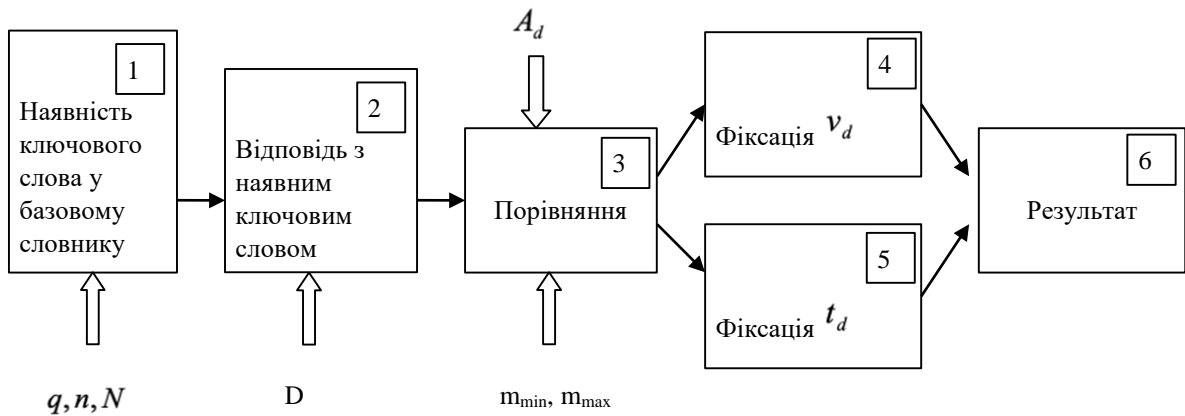


Рисунок 4.8 – Співвіднесення питання, словником базових слів та ключовим словом у відповіді у 3D резюме в процесі проведення експерименту

В імітаційній (робочій) моделі 3D резюме реалізовані наступні блоки системи:

1) базовий словник, сформований, як файл, що складається з N упорядкованих ключових слів за тематикою питань тесту, що мають n -символів з алфавіту q ;

2) Окрім ключових слів, наведені деякі похідні слова у кількості D , які можуть бути використані користувачем для відповіді, де:

a_i – набір слів, аналогічним ключовим $a_1 a_2 \dots a_i$;

\hat{a}_i -поле – частина слів, схожих за значеннями за темою питань $a_1 a_2 \dots a_i$;

$m(\hat{a}_i)$ – кількість слів \hat{a}_i - поля, що використовуються у полі вводу;

$AD^H(\hat{a}_i)$ – початкова адреса (номер першого слова) \hat{a}_i - поля;

$AD^K(\hat{a}_i)$ – кінцева адреса (номер останнього слова) \hat{a}_i - поля;

$A_o^H(\hat{a}_i)$ – слово $a_1 a_2 \dots a_i 00 \dots 0$ (мінімальна кількість введених слів);

$A_0^K(\hat{a}_i)$ – слово $a_1 a_2 \dots a_i (q \cdot 1)(q \cdot 1) \dots (q \cdot 1)$ (максимальна кількість введених слів);

$A[j]$ – співвіднесення слова з номером питання j .

Тобто, для деякого ключового слова A_d , що вводить користувач при відповіді на запитання, є відповідне слово з пакету слів ($d = 1 \dots D$), знаходиться значення $i := 1, 2, \dots$, для якого виконується умова:

$$m_{min} \leq m(\hat{a}_i) \leq m_{max} .$$

Тобто, якщо відповідь складається з $m = 10$ слів, то ключове слово може бути одне або декілька:

$$m_{min} := 1; \quad m_{max} := 10.$$

Якщо відповідь вірна, то достатньо одного ключового слова:

$$m_{min/max} := 1.$$

Процедура розрахунку значення $m(\hat{a}_i) = AD^K(\hat{a}_i) - AD^H(\hat{a}_i) + 1$ наведена в Додатку Ж.

4.3.2 Реалізація підходу прискорення обробки введених слів відповіді

У процесі тестування експериментальної версії 3D резюме було виявлено, що навіть використання стандартних бібліотек та підходів для порівняння відповідності введеного користувачем слова – ключовому, займає багато часу. Для запобігання послідовному перебору всіх слів відповіді та

пошуку відповідності в базовому словнику системи був розроблений ітераційний алгоритм визначення $AD^H(\hat{a}_i), AD^K(\hat{a}_i)$:

Крок 1: Визначити перше наближення $AD_1^H(\hat{a}_i) := AD_0^H(\hat{a}_i) \cdot r$.

Крок 2: Знайти різницю $\Delta_1 = A_0^H(\hat{a}_i) - A(AD_1^H(\hat{a}_i))$:

– якщо $\Delta_1 = 0$, то $A^H(\hat{a}_i) := AD_1^H(\hat{a}_i)$;

– інакше $AD_2^H(\hat{a}_i) := AD_1^H(\hat{a}_i) + \Delta_1$.

Крок 3: Знайти різницю $\Delta_2 = A_0^H(\hat{a}_i) - A(AD_2^H(\hat{a}_i))$:

– якщо $\Delta_2 = 0$, то $A^H(\hat{a}_i) := AD_2^H(\hat{a}_i)$;

– інакше:

а) при $\Delta_2 < 0$, значення $AD_k^H(\hat{a}_i)$ послідовно зменшується на одиницю з обчисленням $\Delta_k = A_0^H(\hat{a}_i) - A(AD_k^H(\hat{a}_i))$. Останнє значення $AD_k^H(\hat{a}_i)$, після якого знак Δ_k змінюється на зворотній, є слово пошуку $AD^H(\hat{a}_i)$;

б) при $\Delta_2 > 0$, значення $AD_k^H(\hat{a}_i)$ послідовно збільшується на одиницю з обчисленням значення Δ_k до зміни знаку Δ_k .

Кроки 1 та 2 – є ітераціями між словами базового словника. Умови кроку 3 є циклами пошуку шляхом відкидання та додавання літер до ключового слова, що є у словнику та порівняння зі словом, що ввів користувач. Якщо таке слово отримане, то воно також заноситься до словника.

Для спрощення алгоритму опущені операції, які пов'язані з коректними обчисленнями значень Δ_k та $AD_k(\hat{a}_i)$.

4.3.3 Результати математичного моделювання та експериментального тестування робочої версії 3D резюме

Аналіз результатів експерименту та математичного моделювання тестування експериментальної версії 3D резюме, дозволив визначити значення дисперсії розподілу сеансних результатів для сигнатур, які характеризуються найбільшим розкидом значень V (сигнатура 4 та 7) мають значення відповідно 0,0081 та 0,0324. Отриманий розподіл близький до розподілу Стюдента. Довірча ймовірність складає $P = 95\%$.

Для сигнатури 4 $\Delta \bar{V}_1^{(3)} = \pm 0,054$; $\bar{V}_1^{(3)} = 2,08$ та $2,026 < V_1^{(3)} < 2,134$, $\Delta V_{\text{відн}} = \frac{0,054}{2,08} 100\% = 2,6\%$, згідно з формулою Чебишова:

$$0,95 < \frac{0,007744}{(\Delta V_1^{(3)})^2}; \Delta V_1^{(3)} < 0,09.$$

Для сигнатури 7 $\Delta \bar{V}_1^{(3)} = \pm 0,11$; $\bar{V}_1^{(3)} = 3,72$ та $3,1 < V_1^{(3)} < 3,83$, $\Delta V_{\text{відн}} = \frac{0,11}{3,72} 100\% \approx 3\%$, згідно з формулою Чебишова:

$$0,95 < \frac{0,0121}{(\Delta V_1^{(3)})^2}; \Delta V_1^{(3)} < 0,18.$$

Отримані значення визначають ступінь статистичної достовірності. У Додатку К наведені та співставлені розрахункові значення $v_1^{(2)}, v_1^{(3)}$, а відповідні їм експериментальні дані для середніх значень V та T .

Тестувалася зручність інтерфейсу при роботі з природною мовою користувача та у випадку роботи з комп'ютерним кодом. Як впливає з таблиці Додатку К, розбіжності між результатами математичного моделювання та експерименту відносно незначні. Для широкого діапазону значень розбіжності не перевищують $N = 10^3 \div 10^6$, відносні розбіжності

лежать в межах $(1 \div 2)\%$, і тільки для однієї сигнатури 8 ($N = 10^2$) розбіжність складає 4%.

Варто підкреслити, що даний приклад лише ілюструє загальний підхід до реалізації 3D резюме у його експериментальному виконанні, бо параметри q , n , N задані стосовно поставленої задачі. При вдосконаленні 3D резюме необхідно буде розширити базовий словник та ускладнити пошук за ключовим словом, наприклад, обмеживши його пошук у першому рядку відповіді респондента з можливістю «навчання» системи розрізняти можливість конкретики при викладенні думки.

Розроблений інтерфейс має дещо високе навантаження інтенсивності праці. Це обумовлено тим, що користувачу в процесі заповнення 3D резюме приходится не лише вибирати відповідь у переліку тестів, а й письмово та у вигляді відео відповідати на деякі завдання. Це вимагає певних зусиль і володіння знаннями та навичками за запитаннями на доволі високому рівні.

4.4 Висновки за розділом 4

В процесі роботи була розроблена експериментальна версія 3D резюме для реалізації підбору, сегментації та навчання кадрів з використанням засобів штучного інтелекту, зокрема, можливістю здійснювати вибір в залежності від відповіді, збереження результатів, порівняння для використання результатів у наступному виборі:

- 1) був реалізований і апробований механізм відтворення людської поведінки при підборі, сегментації та навчанні працівника шляхом комп'ютерного моделювання. Отримана модель дозволила опрацювати механізм переходів між питаннями в залежності від правильної відповіді на попереднє питання, що й було покладено в основу розробки 3D резюме для тестування знань працівника. Розроблене 3D резюме, опрацьований механізм

перебору питань, переходів між окремими вкладками 3D резюме після виконання обов'язкової процедури підтвердження відповідності знань вимогам наявної вакантної посади через аналіз ключових слів у письмовій відповіді респондента;

2) реалізовані процедури заміни номеру питання та тригерів у базі даних 3D резюме. Наведений опис та лістинг кодів реалізації таких процедур при управлінні базою даних 3D резюме;

3) проведене тестування експериментальної версії 3D резюме. Для запобігання послідовному перебору всіх слів відповіді та пошуку відповідності в базовому словнику системи, що може займати багато часу, був розроблений ітераційний алгоритм визначення відповідності введеного слова ключовому слову з базового словника 3D резюме.

Аналіз результатів експерименту та математичного моделювання тестування експериментальної версії 3D резюме дозволив визначити значення дисперсії розподілу сеансних результатів для сигнатур, які характеризуються найбільшим розкидом значень V (сигнатура 4 та 7) мають значення відповідно 0,0081 та 0,0324. Отриманий розподіл близький до розподілу Стюдента. Довірча ймовірність складає $P = 95\%$;

4) матеріали, що подані в розділі, знайшли часткове відображення в роботах [77, 112].

ВИСНОВКИ

За підсумками виконаної роботи можна навести наступні висновки і узагальнення:

1) Проведене узагальнення та систематизація існуючих моделей, методів та підходів при реалізації систем відбору, сегментації та навчання кадрів, з використанням сучасних парадигм кадрового менеджменту дозволило обрати методику дослідження, обґрунтувати мету та задачі, а також представити методологічні основи розробки 3D резюме, як прикладу поєднання елементів штучного інтелекту в інструменті, призначеному для тестуванні знань претендента на роботу. Запропоновано поняття 3D резюме розглядати як HTML-формату, що представляє професійні досягнення особи в трьох вимірах через текст, співбесіду з відеозаписом та тестування в режимі он-лайн.

2) З використанням теорії та математичних основ абстрактних автоматів були запропоновані моделі повтору та вибору для алгоритмізації інтелектуальної діяльності людини при вирішенні логічних завдань або практичних задач, які покладені в основу алгоритму переходу між завданнями за рівнем складності в залежності від результатів відповіді на попередні завдання із запам'ятовуванням отриманого результату. Розроблено метод для використання в розробці інструментальних засобів для відтворення дій людини, що проходить тестування за допомогою мережі Інтернет, коли на початку є деякі вхідні слова (тезаурус спеціальності), а на виході, після складного перебору обмежень, є сформований системою висновок про відповідність даного претендента наведеним професійним вимогам.

3) Представлений логіко-ймовірнісний підхід до формування множини завдань з відповідними алгоритмами реалізації для проведення незалежного оцінювання знань, що забезпечуватимуть унікальність вибору питань для кожного респондента та здійснювати перевірку. Дістали подальшого розвитку підхід Г.П. Донця, В.А. Пепеляєва, О.М. Трофимчука про оптимальне

розбиття на кількість груп з програмною реалізацією стратегії щодо генерування вибору питання для тестування знань працівника. Також отримали подальший розвиток алгоритми за оберненою функцією та за методом Неймана-Пірсона для генерації випадкового номера завдання за заданої аналітичної функції розподілу та алгоритм знаходження відповідності отриманої відповіді за словами-еталонами та ключовим словом завдання.

4) Виконана та протестована експериментальна версія 3D резюме зі здійсненням вибору питання та запам'ятовування результату. Реалізовані процедури заміни номеру питання та тригерів у базі даних 3D резюме. Наведений опис та лістинг кодів реалізації таких процедур при управлінні базою даних 3D резюме. Аналіз результатів експерименту та математичного моделювання за технологією GOMS щодо тестування версії 3D резюме дозволив визначити значення дисперсії розподілу результатів для сигнатур, які характеризуються найбільшим розкидом значень з довірчою ймовірністю 95%.

СПИСОК ВИКОРИСТАНИХ ДЖЕРЕЛ

1. Han, J.; Kamber, M.; Pei, J. (2011), *Data Mining: Concepts and Techniques* (3rd ed.), Morgan Kaufmann, ISBN 978-0-12-381479-1
2. Leavitt, Harold J.; Whisler, Thomas L. (1958). *Management in the 1980s*. *Harvard Business Review*, 11.
3. Бутенко І. Еволюція і генезис основних концепцій управління персоналом. *Вісник економічної науки України*. – 2013 - №2 – сс. 20 – 25.
4. Alan Bundy, Rod Burstall. *Artificial Intelligence: An Introductory Course*. – Revised. – Edinburgh University Press, 1984. – 200 с. – ISBN 978-0852244104.
5. Брінцева О., Біловус О., Інформаційні технології в управлінні персоналом підприємства: сучасні тенденції. *Соціально-трудові відносини: теорія та практика*. Київ. 2018. No 1. С. 264-271.
http://nbuv.gov.ua/UJRN/stvttp_2018_1_28
6. Drucker P.F. *Management : Tasks, Responsibilities, Practices* / P.F. Drucker. – N.Y. Harper and Row, 1973. – 398 p.
7. Логвіненко Б.І. Дослідження інструментів штучного інтелекту в управлінні поведінкою економічних агентів у цифровому просторі на підприємствах. *Вісник Харківського національного університету імені В. Н. Каразіна. Серія «Міжнародні відносини. Економіка. Країнознавство. Туризм»*. – 2022 – Вип. 15. – сс. 45 – 53. ISSN 2310–9513.
8. Fisher, R. (1994). *Introduction to the Position of Independent or Quasi Independent Agencies & Foundations in Cultural Sector in Britain. A Report to the CIRCLE Round Table*. London: Britain Publishers.
9. McConnell, Campbell; Stanley Brue; Sean Flynn; Sean Flynn (2020). *Economics Principles, Problems, & Policies*. McGraw-Hill. 22nd edition. 960 p. ISBN 9781260226775.

10. Duchessi, Peter et al. A Research Perspective: Artificial Intelligence, Management and Organizations. *International Journal of Intelligent Systems in Accounting, Finance & Management* 2 (1993): 151-159.
11. O’Keefe R., O’Leary D. Expert system verification and validation: a survey and tutorial. *Artificial Intelligence Review*. 1993. No 7, pp. 3-42. URL: <https://msbfile03.usc.edu/digitalmeasures/oleary/intellcont/ES-Verification-validation-2.pdf>
12. Kolbjornsrud, V., Amico, R., & Thomas, R. J. Partnering with AI: How organizations can win over skeptical managers. *Strategy and Leadership*. 2017. No 45 (1). pp.37-43.
13. Mounir El Khatib, Ahmed Al Falasi. Effects of Artificial Intelligence on Decision Making in Project Management . *American Journal of Industrial and Business Management* Vol.11 No.3, March 15, 2021. DOI: 10.4236/ajibm.2021.113016
14. O’Leary, D. Validation of Expert Systems, *Decision Sciences*. 1987. No 18- 3, P. 468-486.
15. Harry McGrath, Colin Flanagan, Liaoyuan Zeng, Yiming Lei. Future of Artificial Intelligence in Anesthetics and Pain Management . *Journal of Biosciences and Medicines* Vol.7 No.11, November 13, 2019. DOI: 10.4236/jbm.2019.711010
16. Jupiter Ndiaye, Ousmane Sow, Youssou Traore, Mame Andallah Diop, Ababacar Sadikh Faye, Abdoulaye Diop. Electronic System Using Artificial Intelligence for Queue Management . *Open Journal of Applied Sciences* Vol.12 No.12, December 21, 2022. DOI: 10.4236/ojapps.2022.1212141
17. Baha M. Mohsen. Impact of Artificial Intelligence on Supply Chain Management Performance. *Journal of Service Science and Management* Vol.16 No.1, February 28, 2023. DOI: 10.4236/jssm.2023.161004
18. Брінцева О., Біловус О., Інформаційні технології в управлінні персоналом підприємства: сучасні тенденції. *Соціально-трудова відносинами: теорія та практика*. Київ. 2018. No 1. С. 264-271. http://nbuv.gov.ua/UJRN/stvttp_2018_1_28

19. Матвійчук А. Штучний інтелект в економіці: нейронні мережі, нечітка логіка. Монографія. КНЕУ ім. В. Гетьмана. 2010. С. 361.
20. Піжук О. І. Штучний інтелект як один із ключових драйверів цифрової трансформації економіки. Економіка, управління та адміністрування. 2019. № 3. С. 41-46.
21. Luger, George; Stubblefield, William (2004). Artificial Intelligence: Structures and Strategies for Complex Problem Solving (5th ed.). Benjamin/Cummings. ISBN 978-0-8053-4780-7.
22. Poole, David; Mackworth, Alan; Goebel, Randy (1998). Computational Intelligence: A Logical Approach. New York: Oxford University Press. ISBN 978-0-19-510270-3.
23. Russell, Stuart J.; Norvig, Peter (2003), Artificial Intelligence: A Modern Approach (2nd ed.), Upper Saddle River, New Jersey: Prentice Hall, ISBN 0-13-790395-2.
24. Шевченко А.І., Агарков А.В., Азаренко Д.С., Герасімов І.Г., Дорохіна Г.В., Іванова С.Б., Ніценко А.В., Шелепов В.Ю. Проблеми штучного інтелекту. Аналіз та синтез комунікаційної інформації. Монографія. – /ПШІ МОН України і НАН України/. – Донецьк. ПШІ «Наука і освіта», 2014. – 212 с. (13,4). – Тираж 300. – ISBN 978-966-7829-70-4.
25. Шевченко А.І., Полехін В.Д. Розумова діяльність людини. Монографія.– Донецьк: ПШІ «Наука і освіта», 2007. – 280 с.
26. Шевченко А.И., Звенигородский А.С., Коломыйцев О.А. Критерий оценки графического интерфейса на основе Гештальт принципов. Искусственный интеллект. – № 2. – 2012. – С. 40-45.
27. Шевченко А. И., Курилов М.А., Сыпченко А.П., Барашко А.С. Аксиомы программирования и некоторые вопросы Дистанционного Обучения. Искусственный интеллект. – № 1. – 2012. – С. 251-258.
28. Luthans, F. & Doh Jonathan, P. (2017). International Management, Culture, Strategy and Behavior (10th ed.). Mc Graw Hill. 672 p. · ISBN : 1259705072.

29. Brannan, David A.; Esplen, Matthew F.; Gray, Jeremy J. *Geometry*. Cambridge: Cambridge University Press. 1998. 510 P. ISBN 0-521-59787-0.
30. Прямокутні координати в просторі // Вища математика в прикладах і задачах / Клепко В.Ю., Голець В.Л.. – 2-ге видання. – К.: Центр учбової літератури, 2009. – 594 с.
31. Шевчук С. Українське ділове мовлення : підручник. – К.: Арії, 2007. – 576 с.
32. Drucker, J. (2013). Reading Interface. *PMLA*, 128(1), 213-220. doi:10.1632/pmla.2013.128.1.213
33. Andries van Dam. Post-WIMP User Interfaces // *Communications of the ACM*. – February 1997. – № 40(2). – P. 63-67. – doi:10.1145/253671.253708.
34. Green, Mark; Jacob, Robert (July 1991). "SIGGRAPH '90 Workshop Report: Software Architectures and Metaphors for Non-WIMP User Interfaces". *SIGGRAPH '90*. SIGGRAPH. Dallas: ACM SIGGRAPH. CiteSeerX 10.1.1.121.7982
35. Hsu Y and Schwen T. (2003). The effects of structural cues from multiple metaphors on computer users' information search performance. *International Journal of Human-Computer Studies*. 58:1. (39-55). Online publication date: 1-Jan-2003. [https://doi.org/10.1016/S1071-5819\(02\)00125-8](https://doi.org/10.1016/S1071-5819(02)00125-8)
36. *Ergonomic Requirements for Office Work with Visual Display Terminals*, ISO 9241-11, ISO, Geneva, 1998.
37. Lee, Ju Yeon; Kim, Ju Young; You, Seung Ju; Kim, You Soo; Koo, Hye Yeon; Kim, Jeong Hyun; Kim, Sohye; Park, Jung Ha; Han, Jong Soo; Kil, Siye; Kim, Hyerim (2019-09-30). "Development and Usability of a Life-Logging Behavior Monitoring Application for Obese Patients". *Journal of Obesity & Metabolic Syndrome*. 28 (3): 194–202. doi:10.7570/jomes.2019.28.3.194. ISSN 2508-6235. PMC 6774444. PMID 31583384.
38. Tuch, Alexandre N.; Presslauer, Eva E.; Stöcklin, Markus; Opwis, Klaus; Bargas-Avila, Javier A. (2012-11-01). "The role of visual complexity and prototypicality regarding first impression of websites: Working towards

- understanding aesthetic judgments". *International Journal of Human-Computer Studies*. 70 (11): 794–811. doi:10.1016/j.ijhcs.2012.06.003. ISSN 1071-5819.
39. Bouchard Jr, T. J. Genetic and environmental influences on adult intelligence and special mental abilities // *Human biology; an international record of research : journal*. – 1998. – Vol. 70, no. 2. – P. 257–279. – PMID 9549239
40. Сухий О. Л. Алгоритми пошуку в інформаційних системах : методичні рекомендації / О. Л. Сухий, В. М. Міленін, В. М. Тарадайнік. – К., 2015. – 70с.
41. Майстренко С.Я. Методи та моделі забезпечення достовірності інформації в автоматизованих системах організаційного управління: автореф. дис. на здобуття наук. ступеня канд. техн. наук : спец. 05.13.06 «Інформаційні технології»/ С.Я. Майстренко. – Київ, 2006. – 19 с.
42. Литвинов В.А., Майстренко С.Я., Оксанич І.Н. Трудомісткість інтерфейсу користувача у задачі пошуку взірця в довіднику і її оцінка на основі моделі GOMS / *Математичні машини і системи*. - 2009. - №3. - С. 120-125.
43. Martinez, Wendy L. (23 February 2011). "Graphical user interfaces: Graphical user interfaces". *Wiley Interdisciplinary Reviews: Computational Statistics*. 3 (2): 119–133. doi:10.1002/wics.150. S2CID 60467930.
44. Довгий С. О., Бідюк П. І., Трофимчук О. М., Савенков О. І. Методи прогнозування в системах підтримки прийняття рішень. – К.: Азимут-Україна. – 2011. – 608 с.
45. Довгий С. О., Бідюк П. І., Трофимчук О. М. Системи підтримки прийняття рішень на основі статистично-ймовірнісних методів. – К.: Логос. – 2014. – 419 с.
46. Teorey, T.; Lightstone, S. and Nadeau, T. *Database Modeling & Design: Logical Design*, 4th edition, Morgan Kaufmann Press, 2005. ISBN 0-12-685352-5
47. Halder, Raju; Cortesi, Agostino (2011). "Abstract Interpretation of Database Query Languages" (PDF). *Computer Languages, Systems & Structures*. 38 (2): 123–157. doi:10.1016/j.cl.2011.10.004. ISSN 1477-8424.
48. Gray, Wayne D.; John, Bonnie E.; Atwood, Michael E. (1992). "The Precipice of Project Ernestine or an overview of a validation of GOMS". *Proceedings of the*

- SIGCHI conference on Human factors in computing systems.
doi:10.1145/142750.142821
49. Брінцева О. Г., Білоус О. С. Інформаційні технології в управлінні персоналом підприємства: сучасні тенденції / Соціально-трудові відносини: теорія та практика. – №1. – 2018. – сс. 264 – 271.
50. Грیشнова О.А. Майбутнє праці і працівника в інформаційному суспільстві / Імперативи та інноваційні механізми забезпечення гідної праці в умовах становлення нової економіки : зб. тез доповідей учасників Міжнар. наук.-практ. конф.; 25-26 квіт. 2017 р. – К. : КНЕУ, 2017. – сс. 25 – 26.
51. Дорош, М. С., Грек, І. О., Бугай, Ю. В. (2020). Розробка моделі системи автоматизованого добору персоналу з використанням методів штучного інтелекту. Технічні науки та технології, (2(20)), 158–166. вилучено із <http://tst.stu.cn.ua/article/view/215792>
52. Хміль Ф. Управління персоналом : підручник. – К. : Академвидав, 2006. – 487 с.
53. Бей Г.В., Середа Г.В. Трансформація HR-технологій під впливом цифровізації бізнес-процесів / Економіка і організація управління. – № 2 (34). – 2019. – сс. 93 – 101.
54. Комп'ютерна Академія ШАГ. Електронний ресурс. – режим доступу: <https://kiev.itstep.org/>
55. Глибовець М. М., Олецкий О.В. Штучний інтелект. – Київ : «Києво-Могилянська академія», 2002. – 364 с. – ISBN 966518153X.
56. Життя 3.0. Доба штучного інтелекту / Макс Тегмарк ; пер. з англ. Зорина Корабліна. – Київ: Наш формат, 2019. – 428 с.
57. Баранов С. И. Синтез микропрограммных автоматов. – Л., Энергия. 1979. – 232 с.
58. Moore E. F. Gedanken-experiments on Sequential Machines. Automata Studies, Annals of Mathematical Studies, 1956, 34, 129–153. Princeton University Press, Princeton, N.J.

59. Mealy, George H. (September 1955). "A Method for Synthesizing Sequential Circuits". *Bell System Technical Journal*. 34: 1045–1079. doi:10.1002/j.1538-7305.1955.tb03788.x
60. Прикладна теорія цифрових автоматів. Практикум [Текст] : навч. посіб. для студентів ВНЗ, які навчаються за напрямом підгот. «Безпека інформаційних і комунікаційних систем» / С. М. Головань [та ін.]. – Луганськ : Ноулідж, 2013. – 177 с. : рис., табл. – Бібліогр.: с. 175–176. – 300 прим. – ISBN 978-617-579-540-8.
61. Фрімен Е., Робсон Е. Head First. Програмування на JavaScript. / Пер з англ. Г. Якубовська. – Х.: «Фабула», 2022. – 672 с. ISBN: 978-617-522-047-4.
62. Левин, Р. Практическое введение в технологию искусственного интеллекта и экспертных систем с иллюстрациями на бейсике / Р. Левин, Д. Дранг, Б. Эделсон; Пер. с англ. и предисл. М. Л. Сальникова, Ю. В. Сальниковой. - Москва : Финансы и статистика, 1990. - 237,[2] с. : ил.; 21 см.; ISBN 5-279-00448-0 (В пер.).
63. Електронний парламент України: досвід створення / Довгий С. О., Трофимчук О. М., Лебідь О. Г., Гуляєв К. Д., Кряжич О. О., Сергієнко І. В., Гуляницький Л. Ф., Малишко С. О., Мелашенко А. О., Згуровський М. З., Панкратова Н. Д., Селін Ю. М., Коршевнюк Л. О., Макуха М. М., Савастьянов В. В., Морозов А. О., Кузьменко Г. Є., Баран Л. Б., Вишневський В. В., Коваленко О. Є., Малишев О. В. За ред. чл.-кор. НАНУ, акад. НАПН С. О. Довгого. – К.: Логос. – 2015. – 458 с.
64. Trofymchuk O. M., Vidyuk P.I. Decision support systems, modeling, forecasting, risk estimation. – LAP LAMBERT Academic Publishing – 2019. – 179 p.
65. Trofymchuk, O., Vasyanin, V., Kuzmenko, V. (2016). Complexity of One Packing Optimization Problem. *Cybernetics and Systems Analysis*. 52. 10.1007/s10559-016-9802-9.

66. Гуляєв К. Д., Баховец Е. Б., Гринченко Т. А., Полумиенко С. К., Рыбаков Л. А., Тюрин В. В. Предпосылки становления информационного общества в Украине. Киев, 2008. ISBN 978-966-8405-74-7
67. Копійка О. В. Зміна бізнес-моделі управління ІТ на підприємстві у зв'язку з розвитком сервісно-орієнтованої інформаційної економіки / О. В. Копійка, А. Г. Кондратюк // Математичне моделювання в економіці. - 2018. - № 1. - С. 56-66. - Режим доступу: http://nbuv.gov.ua/UJRN/mmve_2018_1_6.
68. Johnason, P. (2009). HRM in changing organizational contexts. In D. G. Collings & G. Wood (Eds.), Human resource management: A critical approach (pp. 19–37). London: Routledge.
69. Klerck, G. (2009). "Industrial relations and human resource management". In D. G. Collings & G. Wood (Eds.), Human resource management: A critical approach (pp. 238-259). London: Routledge.
70. Armstrong, Michael (2009). Armstrong's handbook of human resource management practice. Armstrong, Michael, 1928- (Eleventh ed.). London: Kogan Page. ISBN 9780749457389. OCLC 435643771.
71. Obedgiu, Vincent (2017-01-01). "Human resource management, historical perspectives, evolution and professional development". Journal of Management Development. 36 (8): 986–990. doi:10.1108/JMD-12-2016-0267
72. Лоусон Б. HTML і XHTML. Детальне керівництво / Б. Лоусон, Р. Шарп. – 2000. – 752 с.
73. Phillips, J. (2004). PMP Project Management Professional Study Guide. p. 354. ISBN 0072230622.
74. Hopcroft, John E., and Jeffrey D. Ullman (2006). Introduction to Automata Theory, Languages, and Computation. 3rd ed Reading, MA: Addison-Wesley. ISBN 978-0-321-45536-9
75. Werner Kluge (2006). Abstract Computing Machines: A Lambda Calculus Perspective. Springer. ISBN 978-3-540-27359-2.
76. Anderson, James A. (2000). Discrete Mathematics with Combinatorics (вид. First). Prentice Hall. с. 799. ISBN 978-0130869982

77. Ющенко К.С. Підхід до автоматизації процесу підбору кадрів за допомогою 3D резюме. Математичні машини і системи. 2022. № 2. С. 29–39.
78. Ющенко К. С. Цифровізація процесів роботи з персоналом / Інформаційно-комунікаційні технології та сталий розвиток // Колективна монографія за матеріалами XXI Міжнародної науково-практичної конференції (Київ, 14-16 листопада 2022 р.) / За заг. ред. С.О. Довгого. – К.: ТОВ «Видавництво «Юстон», 2022. – сс. 206 – 207. ISBN 978-617-7854-76-9.
79. Ющенко К.С. Засоби штучного інтелекту у системах підготовки персоналу // Історія, сучасний стан та тенденції цифрового розвитку суспільства. / Матеріали 10-ої Міжнар. наук.-практ. конф. «Глушковські читання», Київ, 2021 р. / Уклад.: Р.М. Богачев, В.Д. Піхорович, А.Ю. Самарський, М.І. Сторожик. – Київ, 2021. – с. 194 – 196.
80. Гуляев, К. Д., Ющенко, К. С., Купрін, О. М. (2023). Застосування абстрактних автоматів Мілі та Мура для реалізації алгоритмів рекомендації та вибору. International Scientific Technical Journal "Problems of Control and Informatics", 67(6), с. 14–24. doi: 10.34229/1028-0979-2022-6-2. (кат. А)
81. Гуляев К.Д., Каптур В.А., Тіхонов В.І. Принципи організації адаптивної взаємодії відкритих систем /Наукові вісті Національного технічного університету України" Київський політехнічний інститут".-2012.-№ 2.-С. 7-15.-Режим доступу: http://nbuv.gov.ua/UJRN/NVKPI_2012_2_2
82. Довгий С.О., Воробієнко П.П., Гуляев К.Д. Сучасні телекомунікації: Мережі, технології, безпека, економіка, регулювання.–Видання друге (доповнене). - К.:«Азимут-Україна». – 2013.– 608с.
83. Довгий С.О. Інтеграція ІКТ і штучного інтелекту та її вплив на суспільство / С.О. Довгий, К.Д. Гуляев, П.П. Воробієнко // Інфокомунікації – сучасність та майбутнє: матеріали десятої міжнародної науково-практичної конференції, 16-19 листопада 2020 р., м. Одеса. – Одеса: ОНАЗ ім. О.С. Попова, 2020. – С. 16-21. <http://biblio.onat.edu.ua:443/xmlui/handle/123456789/3351>
84. Chakraborty, P.; Saxena, P. C.; Katti, C. P. (2011). Fifty Years of Automata Simulation: A Review. ACM Inroads. 2 (4): 59–70. doi:10.1145/2038876.2038893

85. Ferraguti, A.; Micheli, G.; Schnyder, R. (2018), Irreducible compositions of degree two polynomials over finite fields have regular structure, *The Quarterly Journal of Mathematics*, vol. 69, Oxford University Press, pp. 1089–1099, doi:10.1093/qmath/hay015
86. Issel, W. (1979). Aho, A. V. / Hopcroft, J. E. / Ullman, J. D., *The Design and Analysis of Computer Algorithms*. London-Amsterdam-Don Mills-Sydney. Addison-Wesley Publ. Comp. 1974 X, 470 S., \$ 24,-. *ZAMM - Zeitschrift für Angewandte Mathematik und Mechanik* (нім.) 59 (2). с. 141–141. doi:10.1002/zamm.19790590233.
87. Hromkovič, Juraj, (2004). *Theoretical computer science: introduction to Automata, computability, complexity, algorithmics, randomization, communication, and cryptography*. Berlin: Springer. ISBN 3540140158.
88. Dale, Sarah (2015). Heuristics and biases: The science of decision-making. *Business Information Review*. 32 (2): 93–99. doi:10.1177/0266382115592536
89. Braun, T.D.; Siegal, H.J.; Beck, N.; Boloni, L.L.; Maheswaran, M.; Reuther, A.I.; Robertson, J.P.; Theys, M.D.; Bin Yao; Hensgen, D.; Freund, R.F. (1999). A comparison study of static mapping heuristics for a class of meta-tasks on heterogeneous computing systems. *Proceedings. Eighth Heterogeneous Computing Workshop (HCW'99)*. IEEE Comput. Soc: 15–29. doi:10.1109/hcw.1999.765093
90. arnaugh, Maurice (November 1953) [1953-04-23, 1953-03-17]. "The Map Method for Synthesis of Combinational Logic Circuits" (PDF). *Transactions of the American Institute of Electrical Engineers, Part I: Communication and Electronics*. 72 (5): 593–599. doi:10.1109/TCE.1953.6371932. Paper 53-217.
91. Brown, Frank Markham (2012). *Boolean Reasoning - The Logic of Boolean Equations* (reissue of 2nd ed.). Mineola, New York: Dover Publications, Inc. ISBN 978-0-486-42785-0.
92. Farhat, Hassan A. (2004). *Digital design and computer organization*. Vol. 1. CRC Press. p. 274. ISBN 978-0-8493-1191-8.
93. Al-hayanni, Mohammed A. Noaman; Xia, Fei; Rafiev, Ashur; Romanovsky, Alexander; Shafik, Rishad; Yakovlev, Alex (2020). Amdahl's law in the context of

- heterogeneous many-core systems – a survey. *IET Computers & Digital Techniques*. 14 (4): 133–148. doi:10.1049/iet-cdt.2018.5220
94. Gentle, J.E. (2009). *Computational Statistics*. Springer. ISBN 978-0-387-98145-1.
95. Кряжич О.О., Коваленко О.В. Забезпечення алгоритмічного та ресурсного балансу при вирішенні ситуаційних задач. *Математичне моделювання в економіці*. – 2015. - №1. – С. 22-31.
96. Kryazhych O., Itskovych V., Iushchenko K., Hrytsyshyna V., Bruvier D., Nykytyuk V., Bodnarchuk I. (2023) The use of abstract moore automaton to control the sensors of a service-oriented alarm and emergency notification network. *Scientific Journal of TNTU (Tern.)*, vol 109, no 1, pp. 111–120. (кат. Б)
97. Гавриленко О. В. Огляд методів гібридизації інтелектуальної обробки інформації / О. В. Гавриленко // *Науковий вісник Херсонського державного університету. Сер. : Економічні науки*. - 2014. - Вип. 8(1). - С. 204-207. - Режим доступу: [http://nbuv.gov.ua/UJRN/Nvkhdu_en_2014_8\(1\)_51](http://nbuv.gov.ua/UJRN/Nvkhdu_en_2014_8(1)_51).
98. Skakalina E.V. Hybridization of the genetic algorithm with the apparatus of fuzzy sets [Electronic resource] / E.V. Skakalina // *Proceedings of Fourth International Scientific and Technical Conference "Computer and information systems and technologies"*, April 22-23, 2020. – Kharkov : NURE. 2020. – URL : <http://csitic.nure.ua/issue/view/12193/showToc>
99. Luger G. F. *Artificial Intelligence: Structures and Strategies for Complex Problem Solving*. Addison Wesley; 5th edition. 2004. P. 928. ISBN: 978-0321263186
100. Комбінаторні алгоритми підтримки прийняття управлінських рішень / Г.П. Донець, В.А. Пепеляєв, О.М. Трофимчук // *Доповіді Національної академії наук України*. – 2014. – № 11. – С. 33-39. – Бібліогр.: 6 назв. – укр.
101. David E. Bock, Paul F. Velleman, Richard D. De Veaux (2007). "Stats, Modeling the World," pp. 606-627, Pearson Addison Wesley, Boston, ISBN 0-13-187621-X

102. Lehmann E.L., Romano J. P. Testing statistical hypotheses. New York: Springer, 2008. 786 p.
103. Thomason, R. (2020). Logic and Artificial Intelligence. The Stanford Encyclopedia of Philosophy. Edward N. Zalta (ed.). <https://plato.stanford.edu/archives/sum2020/entries/logic-ai/>
104. From Standard Logic to Logic Programming: Introducing a Logic Based Approach to Artificial Intelligence by Gochet, Paul; Grégoire, Eric; Gribomont, Pascal; Hulin, Guy; Pirotte, Alain; Roelants, Dominique; Snyers, Dominique; Thayse, André; Vauclair, Marc; Wolper, Pierre. Wiley, 1988. 1st Edition . P.432. ISBN: 9780471918387
105. Карташов М. В. Імовірність, процеси, статистика. – Київ : ВПЦ Київський університет, 2007. – 504 с.
106. Guoping Zeng (June 2003), "Two common properties of the erlang-B function, erlang-C function, and Engset blocking function", Mathematical and Computer Modelling, Elsevier Science, 37 (12–13): 1287–1296, doi:10.1016/S0895-7177(03)90040-9
107. Rolski, Tomasz; Schmidli, Hanspeter; Schmidt, Volker; Teugels, Jozef (2008). "Risk Processes". Stochastic Processes for Insurance & Finance. Wiley Series in Probability and Statistics. pp. 147–204. doi:10.1002/9780470317044.ch5
108. Кряжич О.О., Ющенко К.С., Іцкович В.Є., Купрін О.М. Особливості алгоритмізації процесів мінімізації похибок апроксимації при вирішенні прикладних задач. Математичні машини і системи. – 2023 - №1 – сс. 118 – 129. (кат. Б)
109. Kryazhych, O., Itskovych, V., Iushchenko, K., Kuprin, O. (2023). Features in solving individual tasks to develop service-oriented networks using dynamic programming. Eastern-European Journal of Enterprise Technologies, 1 (4 (121)), 34–40. doi: <https://doi.org/10.15587/1729-4061.2023>. (Scopus, Q3)
110. Purdum, Jack (2015). Beginning C for Arduino: Learn C Programming for the Arduino (2nd ed.). Apress. ISBN 978-1484209417.

111. Офіційна сторінка онлайн-програми для 3D-моделювання *Tinkercad*
<https://www.tinkercad.com/dashboard>
112. Kryazhych, O., Kovalenko, O., Itskovych, V., & Iushchenko, K. (2023). The modeling of changes in the specific activity of tritium in plants. *EUREKA: Physics and Engineering*, (3), 3-14. <https://doi.org/10.21303/2461-4262.2023.002877> (Scopus, Q2)

ДОДАТКИ

ДОДАТОК А

Акти впровадження результатів дисертаційної роботи

«Затверджую»

Товариство з обмеженою
відповідальністю «Будівельно-проектна
фірма Україна»

Ляшук К.В.



АКТ

впровадження розробки за результатами дисертаційної роботи на здобуття ступеня доктора філософії за темою «Моделі та інструментальні засоби для прийняття рішень за умов невизначеності при автоматизованому відборі персоналу на Інтернет-платформах» *Ющенко Катерини Сергіївни*

Цей АКТ складено про те, що методологічні підходи до автоматизованого вибору питань та алгоритмічні рішення для тестування знань та вмінь працівників, що були розроблені в процесі виконання дисертаційної роботи на здобуття ступеня доктора філософії за темою «Використання засобів штучного інтелекту в сучасних системах підбору, сегментації та навчання кадрів» Ющенко Катерини Сергіївни, були впроваджені в роботу Товариства з обмеженою відповідальністю «Будівельно-проектна фірма Україна» для створення форми навчання кадрів з техніки безпеки при виконанні будівельних робіт.

Директор

Інженер з охорони праці



Ляшук К.В.

Микитюк О.В.

Федерация професійних спілок України

Federation of Trade Unions of Ukraine

АКАДЕМІЯ ПРАЦІ,
СОЦІАЛЬНИХ ВІДНОСИН
І ТУРИЗМУ
(АПСВТ)ЄДРПОУ: 04641405
Кільцева дорога, 3-А
м. Київ, 03188 Україна
тел./факс: (044) 522-49-40
email: info@socosvita.kiev.uaACADEMY OF LABOUR,
SOCIAL RELATIONS
AND TOURISM
(ALRST)USREOU: 04641405
3-A, Kiltseva Rd
Kyiv, 03188 Ukraine
tel./fax: +38 (044) 522-49-40
email: info@socosvita.kiev.ua

17.09.2023 № 01/02-438

На № _____ від _____ **АКТ**

впровадження результатів дисертаційного дослідження Ющенко Катерини Сергіївни на тему: «Моделі та інструментальні засоби для прийняття рішень за умов невизначеності при автоматизованому відборі персоналу на Інтернет-платформах» на здобуття наукового ступеня доктора філософії зі спеціальності «Комп'ютерні науки»

Академією праці, соціальних відносин і туризму в 2023-2024 рр. здійснювалась апробація та впровадження результатів дисертаційного дослідження Ющенко Катерини Сергіївни на тему: «Моделі та інструментальні засоби для прийняття рішень за умов невизначеності при автоматизованому відборі персоналу на Інтернет-платформах». Зокрема, під час підготовки навчально методичного забезпечення навчальних дисциплін, таких як «Використання технологій штучного інтелекту у фінансах», «Fintech: трансформація у фінансових сервісах» та «Продакт менеджмент» використовувалися матеріали дисертаційного дослідження щодо застосування алгоритму аналітичної функції розподілу тестових питань, з переходами та запам'ятовуванням відповіді для проведення порівняння набутих знань та навичок студентів.

Методологічні підходи розроблені автором щодо використання засобів штучного інтелекту в сучасних системах підбору, сегментації та навчання кадрів впроваджені при розробці конспектів лекцій для студентів Академії праці, соціальних відносин і туризму для спеціальності фінанси, банківська справа та страхування.

Використання зазначених розробок дозволяє більш кваліфіковано підготувати сучасних фахівців у сфері управління та адміністрування.

Акт впровадження результатів дисертаційного дослідження Ющенко Катерини Сергіївни на тему: «Моделі та інструментальні засоби для прийняття рішень за умов невизначеності при автоматизованому відборі персоналу на Інтернет-платформах» обговорено та схвалено на засіданні кафедри фінансів (протокол № 13 від 21.06.2023р.).

Завідувач кафедри фінансів
Декан факультету економіки,
соціальних технологій та туризму
Проректор з навчально-педагогічної
роботи

Я.С. Ткаченко

С.В. Шолудченко

І.С. Чорнодід

00000194

**Алгоритм виконання розрахунків матриці переходів автомату М для
застосування евристичного методу кодування станів**

Крок 1.

| | |
|-----|-----|
| M= | 1-5 |
| | 1-1 |
| | 1-7 |
| | 2-1 |
| | 2-3 |
| | 2-4 |
| | 3-6 |
| | 3-3 |
| | 3-5 |
| | 4-4 |
| | 4-6 |
| | 4-2 |
| | 5-7 |
| | 5-1 |
| | 5-8 |
| | 6-5 |
| | 6-9 |
| | 6-8 |
| | 7-4 |
| | 7-8 |
| 7-1 | |
| 8-9 | |
| 8-1 | |
| 8-3 | |
| 9-7 | |
| 9-9 | |
| 9-1 | |

У матриці М викреслюємо перший рядок (1-5) і ті рядки, які виявилися повністю закодованими (1-1, 5-1).

$$M' = \begin{array}{|l} 1-7 \\ 2-3 \\ 2-1 \\ 2-4 \\ 3-6 \\ 3-3 \\ 3-5 \\ 4-4 \\ 4-6 \\ 4-2 \\ 5-7 \\ 5-8 \\ 6-5 \\ 6-9 \\ 6-8 \\ 7-4 \\ 7-8 \\ 7-1 \\ 8-9 \\ 8-1 \\ 8-3 \\ 9-7 \\ 9-9 \\ 9-1 \end{array}$$

Отримуємо матрицю M' .

В матриці M' в першому рядку не закодований елемент 7: $\gamma=7$. Будуємо матрицю M_γ (M_7), вибравши з матриці M' все рядки, що містять $\gamma = 7$.

$$M_7 = \begin{array}{|l} 1-7 \\ 5-7 \\ 7-4 \\ 7-8 \\ 7-1 \\ 9-7 \end{array}$$

$B_7 = \{1, 5\} = \{0000, 0001\}$ – множина елементів з матриці M_7 , що вже закодовані.

Для кожного елемента цієї множини знаходимо множини:

$$C_{\gamma f}^1 = C_{71}^1 = \{0010, 0100, 1000\};$$

$$C_{\gamma f}^1 = C_{75}^1 = \{0011, 0101, 1001\};$$

$$D_7^1 = \{0010, 0100, 1000, 0011, 0101, 1001\}.$$

Для кожного коду з множини D_7^1 знайдем $w_{gf} = |k_{\delta g} - k_{\gamma f}|^2$ – межкодова відстань за Хеммінгом, де: $W_g = \sum_{f=1}^F W_{gf}$.

$$W_{0010} = \begin{array}{c} 1-7 \\ \left| \begin{array}{c} 0000 \\ 0010 \end{array} \right| + \begin{array}{c} 5-7 \quad 7-1 \\ \left| \begin{array}{c} 0001 \\ 0010 \end{array} \right| + \left| \begin{array}{c} 0010 \\ 0000 \end{array} \right| \end{array} = 4$$

$$W_{0100} = \begin{array}{c} \left| \begin{array}{c} 0000 \\ 0100 \end{array} \right| + \left| \begin{array}{c} 0001 \\ 0100 \end{array} \right| + \left| \begin{array}{c} 0100 \\ 0000 \end{array} \right| \end{array} = 4$$

$$W_{1000} = \begin{array}{c} \left| \begin{array}{c} 0000 \\ 1000 \end{array} \right| + \left| \begin{array}{c} 0001 \\ 1000 \end{array} \right| + \left| \begin{array}{c} 1000 \\ 0000 \end{array} \right| \end{array} = 4$$

$$W_{0011} = \begin{array}{c} \left| \begin{array}{c} 0000 \\ 0011 \end{array} \right| + \left| \begin{array}{c} 0001 \\ 0011 \end{array} \right| + \left| \begin{array}{c} 0011 \\ 0000 \end{array} \right| \end{array} = 5$$

$$W_{0101} = \begin{array}{c} \left| \begin{array}{c} 0000 \\ 0101 \end{array} \right| + \left| \begin{array}{c} 0001 \\ 0101 \end{array} \right| + \left| \begin{array}{c} 0101 \\ 0000 \end{array} \right| \end{array} = 5$$

$$W_{1001} = \begin{array}{c} \left| \begin{array}{c} 0000 \\ 1001 \end{array} \right| + \left| \begin{array}{c} 0001 \\ 1001 \end{array} \right| + \left| \begin{array}{c} 1001 \\ 0000 \end{array} \right| \end{array} = 5$$

Вибираємо код, для якого w_g має мінімальне значення:

$K_7=0010$.

Крок 2.

З матриці M викреслюємо перший рядок (1-7) і ті рядки, що виявились повністю закодованими (5-7, 7-1). В результаті отримуємо нову матрицю M' , в першому рядку якої не закодований елемент 2: $\gamma=2$. Будується матриця M_2 (M_2), шляхом вибору з матриці M' всі строки, що містять γ (2).

$$M' = \begin{array}{|l} 2-1 \\ 2-3 \\ 2-4 \\ 3-6 \\ 3-3 \\ 3-5 \\ 4-4 \\ 4-6 \\ 4-2 \\ 5-8 \\ 6-5 \\ 6-9 \\ 6-8 \\ 7-4 \\ 7-8 \\ 8-9 \\ 8-1 \\ 8-3 \\ 9-7 \\ 9-9 \\ 9-1 \end{array}$$

$$M_2 = \begin{array}{|l} 2-1 \\ 2-3 \\ 2-4 \\ 4-2 \end{array}$$

$B_2 = \{1\} = \{0000\}$ – множина елементів з матриці M_2 , які вже закодовані.

Знаходимо множини

$$C_{\gamma f}^1 = C_{21}^1 = \{0100, 1000\};$$

$$D_2^1 = \{0100, 1000\};$$

Для кожного коду з множини D_2^1 знаходимо

$w_{gf} = |k_{\delta g} - k_{\gamma f}|^2$ – міжкодова відстань за Хеммінгом.

$$W_{0100} = \begin{vmatrix} 0100 \\ 0000 \end{vmatrix} = 1$$

$$W_{1000} = \begin{vmatrix} 1000 \\ 0000 \end{vmatrix} = 1$$

Обираємо $K_2 = 0100$.

З матриці M' викреслюємо перший рядок (2-1). В результаті отримуємо нову матрицю M' , в першому рядку якої не закодований елемент 3: $\gamma=3$. Будуємо матрицю $M_\gamma (M_3)$, вибравши з матриці M' всі строки, що містять $\gamma=3$.

$$M = \begin{array}{|c} 2-3 \\ 2-4 \\ 3-6 \\ 3-3 \\ 3-5 \\ 4-4 \\ 4-6 \\ 4-2 \\ 5-8 \\ 6-5 \\ 6-9 \\ 6-8 \\ 7-4 \\ 7-8 \\ 8-9 \\ 8-1 \\ 8-3 \\ 9-7 \\ 9-9 \\ 9-1 \end{array}$$

$$M_3 = \begin{array}{|c} 2-3 \\ 3-6 \\ 3-3 \\ 3-5 \\ 8-3 \end{array}$$

$V_3 = \{2, 5\} = \{0100, 0001\}$ – множина елементів з матриці M_3 , котрі вже закодовані. Знаходимо множини:

$$C_{\gamma f}^1 = C_{32}^1 = \{0101, 0110, 1100\};$$

$$C_{\gamma f}^1 = C_{35}^1 = \{1001, 1010, 1100\};$$

$$D_3^1 = \{0101, 0110, 1100, 1001\};$$

Для кожного коду з множини D_3^1 знаходимо

$w_{gf} = |k_{\delta g} - k_{\gamma f}|^2$ – міжкодова відстань по Хемінгу.

$$W_{0101} = \begin{array}{c} 2-3 \\ \left| \begin{array}{c} 0100 \\ 0101 \end{array} \right| \end{array} + \begin{array}{c} 3-3 \\ \left| \begin{array}{c} 0101 \\ 0101 \end{array} \right| \end{array} + \begin{array}{c} 3-5 \\ \left| \begin{array}{c} 0101 \\ 0001 \end{array} \right| \end{array} = 2$$

$$W_{0110} = \begin{array}{c} \left| \begin{array}{c} 0100 \\ 0110 \end{array} \right| \end{array} + \begin{array}{c} \left| \begin{array}{c} 0110 \\ 0110 \end{array} \right| \end{array} + \begin{array}{c} \left| \begin{array}{c} 0110 \\ 0001 \end{array} \right| \end{array} = 4$$

$$W_{1100} = \begin{array}{c} \left| \begin{array}{c} 0100 \\ 1100 \end{array} \right| \end{array} + \begin{array}{c} \left| \begin{array}{c} 1100 \\ 1100 \end{array} \right| \end{array} + \begin{array}{c} \left| \begin{array}{c} 1100 \\ 0001 \end{array} \right| \end{array} = 4$$

$$W_{1010} = \begin{array}{c} \left| \begin{array}{c} 0100 \\ 1010 \end{array} \right| \end{array} + \begin{array}{c} \left| \begin{array}{c} 1010 \\ 1010 \end{array} \right| \end{array} + \begin{array}{c} \left| \begin{array}{c} 1010 \\ 0001 \end{array} \right| \end{array} = 6$$

Вибираємо код з мінімальним значенням W .

$K_3=0101$.

Крок 3.

З матриці M' викреслюємо перший рядок (2-3) і ті рядки, які виявилися повністю закодованими (3-3, 3-5). В результаті отримуємо нову матрицю M'' , в першому рядку якої не закодований елемент 4: $\gamma=4$. Будуємо матрицю $M\gamma$ (M_4), вибравши з матриці M' всі рядки, що містять $\gamma = 4$.

$$M' = \begin{array}{|c} 2-4 \\ 3-6 \\ 4-4 \\ 4-6 \\ 4-2 \\ 5-8 \\ 6-5 \\ 6-9 \\ 6-8 \\ 7-4 \\ 7-8 \\ 8-9 \\ 8-1 \\ 8-3 \\ 9-7 \\ 9-9 \\ 9-1 \end{array}$$

$$M_4 = \begin{array}{|c} 2-4 \\ 4-4 \\ 4-6 \\ 4-2 \\ 7-4 \end{array}$$

$B_4 = \{2, 7\} = \{0100, 0010\}$ – множина елементів з матриці M_4 , які вже закодовані. Знаходимо множини

$$C_{\gamma f}^1 = C_{42}^1 = \{0110, 1100\};$$

$$C_{\gamma f}^1 = C_{47}^1 = \{0011, 0110, 1010\};$$

$$D_4^1 = \{0110, 1100, 0011, 1010\};$$

Для кожного коду з множини D_4^1 знаходимо

$$w_{gf} = |k_{\delta g} - k_{\gamma f}|^2 - \text{міжкодова відстань за Хеммінгом.}$$

$$W_{0110} = \begin{matrix} 2-4 \\ \left| \begin{array}{c} 0100 \\ 0110 \end{array} \right| \end{matrix} + \begin{matrix} 4-4 \\ \left| \begin{array}{c} 0110 \\ 0110 \end{array} \right| \end{matrix} + \begin{matrix} 4-2 \\ \left| \begin{array}{c} 0110 \\ 0100 \end{array} \right| \end{matrix} + \begin{matrix} 7-4 \\ \left| \begin{array}{c} 0010 \\ 0110 \end{array} \right| \end{matrix} = 3$$

$$W_{1100} = \begin{matrix} \left| \begin{array}{c} 0100 \\ 1100 \end{array} \right| \end{matrix} + \begin{matrix} \left| \begin{array}{c} 1100 \\ 1100 \end{array} \right| \end{matrix} + \begin{matrix} \left| \begin{array}{c} 1100 \\ 0100 \end{array} \right| \end{matrix} + \begin{matrix} \left| \begin{array}{c} 0010 \\ 1100 \end{array} \right| \end{matrix} = 5$$

$$W_{0011} = \begin{matrix} \left| \begin{array}{c} 0100 \\ 0011 \end{array} \right| \end{matrix} + \begin{matrix} \left| \begin{array}{c} 0011 \\ 0011 \end{array} \right| \end{matrix} + \begin{matrix} \left| \begin{array}{c} 0011 \\ 0100 \end{array} \right| \end{matrix} + \begin{matrix} \left| \begin{array}{c} 0010 \\ 0011 \end{array} \right| \end{matrix} = 7$$

$$W_{1010} = \begin{matrix} \left| \begin{array}{c} 0100 \\ 1010 \end{array} \right| \end{matrix} + \begin{matrix} \left| \begin{array}{c} 1010 \\ 1010 \end{array} \right| \end{matrix} + \begin{matrix} \left| \begin{array}{c} 1010 \\ 0100 \end{array} \right| \end{matrix} + \begin{matrix} \left| \begin{array}{c} 0010 \\ 1010 \end{array} \right| \end{matrix} = 7$$

Вибираємо код з мінімальним значенням W .

$K_4=0110$.

Крок 4.

З матриці M' викреслимо перший рядок (2-4) і ті рядки, які виявилися повністю закодованими (4-4, 4-2, 7-4). В результаті отримуємо нову матрицю M' , в першому рядку якої не закодований елемент б: $\gamma=6$. Будуємо матрицю $M_\gamma (M_6)$, вибравши з матриці M' всі рядки, що містять $\gamma = 6$.

$$M' = \begin{array}{|c} 3-6 \\ 4-6 \\ 5-8 \\ 6-5 \\ 6-9 \\ 6-8 \\ 7-8 \\ 8-9 \\ 8-1 \\ 8-3 \\ 9-7 \\ 9-9 \\ 9-1 \end{array}$$

$$M_6 = \begin{array}{|c} 3-6 \\ 4-6 \\ 6-5 \\ 6-9 \\ 6-8 \end{array}$$

$B_6 = \{3, 4, 5\} = \{0101, 0110, 0001\}$ – множина елементів з матриці M_6 , які вже закодовані. Знаходимо множини:

$$C_{\gamma f}^1 = C_{63}^1 = \{1101, 0111\};$$

$$C_{\gamma f}^1 = C_{64}^1 = \{1110, 0111\};$$

$$C_{\gamma f}^1 = C_{65}^1 = \{1001, 0011\};$$

$$D_6^1 = \{1101, 0111, 1110, 1001, 0011\};$$

Для кожного коду з множини D_6^1 знаходимо

$w_{gf} = |k_{\delta g} - k_{\gamma f}|^2$ – міжкодова відстань за Хеммінгом.

$$W_{1101} = \begin{array}{|c} 3-6 & 4-6 & & 6-5 \end{array} \begin{array}{|c} 0101 \\ 1101 \end{array} + \begin{array}{|c} 0110 \\ 1101 \end{array} + \begin{array}{|c} 1101 \\ 0001 \end{array} = 6$$

$$W_{0111} = \begin{array}{|c} 0101 \\ 0111 \end{array} + \begin{array}{|c} 0110 \\ 0111 \end{array} + \begin{array}{|c} 0111 \\ 0001 \end{array} = 4$$

$$W_{1110} = \begin{vmatrix} 0101 \\ 1110 \end{vmatrix} + \begin{vmatrix} 0110 \\ 1110 \end{vmatrix} + \begin{vmatrix} 1110 \\ 0001 \end{vmatrix} = 8$$

$$W_{1001} = \begin{vmatrix} 0101 \\ 1001 \end{vmatrix} + \begin{vmatrix} 0110 \\ 1001 \end{vmatrix} + \begin{vmatrix} 1001 \\ 0001 \end{vmatrix} = 7$$

$$W_{0011} = \begin{vmatrix} 0101 \\ 0011 \end{vmatrix} + \begin{vmatrix} 0110 \\ 0011 \end{vmatrix} + \begin{vmatrix} 0011 \\ 0001 \end{vmatrix} = 5$$

Обираємо код з мінімальним значенням W .

$K_6=0111$.

Крок 5.

З матриці M' викреслюємо перший рядок (3-6) і ті рядки, які виявилися повністю закодованими (4-6, 6-5). В підсумку отримуємо нову матрицю M' , у першому рядку якої залишається незакодованим елемент 8: $\gamma=8$. Будуємо матрицю M_γ (M_8), обираючи з матриці M' всі рядки, що мають $\gamma = 8$.

$$M' = \begin{vmatrix} 5-8 \\ 6-9 \\ 6-8 \\ 7-8 \\ 8-9 \\ 8-1 \\ 8-3 \\ 9-7 \\ 9-9 \\ 9-1 \end{vmatrix}$$

$$M_8 = \begin{array}{|c} 5-8 \\ 6-8 \\ 7-8 \\ 8-9 \\ 8-1 \\ 8-3 \end{array}$$

$B_8 = \{5, 6, 7, 1, 3\} = \{0001, 0111, 0010, 0000, 0101\}$ – множина елементів з матриці M_8 , які вже закодовані. Знаходимо множини:

$$C_{\gamma f}^1 = C_{85}^1 = \{1001, 0011\};$$

$$C_{\gamma f}^1 = C_{86}^1 = \{1111, 0011\};$$

$$C_{\gamma f}^1 = C_{87}^1 = \{1010, 0011\};$$

$$C_{\gamma f}^1 = C_{81}^1 = \{1000\};$$

$$C_{\gamma f}^1 = C_{83}^1 = \{1101\};$$

$$D_8^1 = \{1001, 0011, 1111, 1010, 1000, 1101\};$$

Для кожного коду з множини D_8^1 знаходимо $w_{gf} = |k_{\delta g} - k_{\gamma f}|^2$ – міжкодова відстань за Хеммінгом.

$$W_{1001} = \begin{array}{c} 5-8 \\ \left| \begin{array}{c} 0001 \\ 1001 \end{array} \right| \end{array} + \begin{array}{c} 6-8 \\ \left| \begin{array}{c} 0111 \\ 1001 \end{array} \right| \end{array} + \begin{array}{c} 7-8 \\ \left| \begin{array}{c} 0010 \\ 1001 \end{array} \right| \end{array} + \begin{array}{c} 8-1 \\ \left| \begin{array}{c} 1001 \\ 0000 \end{array} \right| \end{array} + \begin{array}{c} 8-3 \\ \left| \begin{array}{c} 1001 \\ 0101 \end{array} \right| \end{array} = 10$$

$$W_{0011} = \begin{array}{c} \left| \begin{array}{c} 0001 \\ 0011 \end{array} \right| \end{array} + \begin{array}{c} \left| \begin{array}{c} 0111 \\ 0011 \end{array} \right| \end{array} + \begin{array}{c} \left| \begin{array}{c} 0010 \\ 0011 \end{array} \right| \end{array} + \begin{array}{c} \left| \begin{array}{c} 0011 \\ 0000 \end{array} \right| \end{array} + \begin{array}{c} \left| \begin{array}{c} 0011 \\ 0101 \end{array} \right| \end{array} = 7$$

$$W_{1111} = \begin{array}{c} \left| \begin{array}{c} 0001 \\ 1111 \end{array} \right| \end{array} + \begin{array}{c} \left| \begin{array}{c} 0111 \\ 1111 \end{array} \right| \end{array} + \begin{array}{c} \left| \begin{array}{c} 0010 \\ 1111 \end{array} \right| \end{array} + \begin{array}{c} \left| \begin{array}{c} 1111 \\ 0000 \end{array} \right| \end{array} + \begin{array}{c} \left| \begin{array}{c} 1111 \\ 0101 \end{array} \right| \end{array} = 13$$

$$W_{1010} = \begin{vmatrix} 0001 \\ 1010 \end{vmatrix} + \begin{vmatrix} 0111 \\ 1010 \end{vmatrix} + \begin{vmatrix} 0010 \\ 1010 \end{vmatrix} + \begin{vmatrix} 1010 \\ 0000 \end{vmatrix} + \begin{vmatrix} 1010 \\ 0101 \end{vmatrix} = 13$$

$$W_{1000} = \begin{vmatrix} 0001 \\ 1000 \end{vmatrix} + \begin{vmatrix} 0111 \\ 1000 \end{vmatrix} + \begin{vmatrix} 0010 \\ 1000 \end{vmatrix} + \begin{vmatrix} 1000 \\ 0000 \end{vmatrix} + \begin{vmatrix} 1000 \\ 0101 \end{vmatrix} = 12$$

$$W_{1101} = \begin{vmatrix} 0001 \\ 1101 \end{vmatrix} + \begin{vmatrix} 0111 \\ 1101 \end{vmatrix} + \begin{vmatrix} 0010 \\ 1101 \end{vmatrix} + \begin{vmatrix} 1101 \\ 0000 \end{vmatrix} + \begin{vmatrix} 1101 \\ 0101 \end{vmatrix} = 12$$

Обираємо код з мінімальним значенням W .

$K_6=0011$.

Крок 6.

З матриці M викреслюємо перший рядок (5-8) і ті рядки, які виявилися повністю закодованими (6-8, 7-8, 8-1, 8-3). В підсумку отримуємо нову матрицю M' , в першому рядку якої є незакодований елемент 9. $\gamma=9$. Будуємо матрицю M_γ (M_9), обираючи з матриці M' всі рядки, що мають $\gamma = 9$.

$$M' = \begin{vmatrix} 6-9 \\ 8-9 \\ 9-7 \\ 9-9 \\ 9-1 \end{vmatrix}$$

$$M_9 = \begin{vmatrix} 6-9 \\ 8-9 \\ 9-7 \\ 9-9 \\ 9-1 \end{vmatrix}$$

$B_9 = \{6, 8, 7, 1\} = \{0111, 0011, 0010, 0000\}$ – множина елементів з матриці

M_9 , які вже закодовані. Знаходимо множини:

$$C_{\gamma f}^1 = C_{96}^1 = \{1111\};$$

$$C_{\gamma f}^1 = C_{98}^1 = \{1011\};$$

$$C_{\gamma f}^1 = C_{97}^1 = \{1010\};$$

$$C_{\gamma f}^1 = C_{91}^1 = \{1000\};$$

$$D_9^1 = \{1111, 1011, 1010, 1000\}.$$

Для кожного коду з множини D_9^1 знаходимо $w_{gf} = |K_{\delta g} - K_{\gamma f}|^2$ – міжкодову відстань за Хеммінгом.

$$W_{1111} = \begin{array}{c} 6-9 \quad 8-9 \\ \left| \begin{array}{c} 0111 \\ 1111 \end{array} \right| + \begin{array}{c} 9-7 \\ \left| \begin{array}{c} 0011 \\ 1111 \end{array} \right| + \begin{array}{c} 9-9 \\ \left| \begin{array}{c} 1111 \\ 0010 \end{array} \right| + \begin{array}{c} 9-1 \\ \left| \begin{array}{c} 1111 \\ 1111 \end{array} \right| + \left| \begin{array}{c} 1111 \\ 0000 \end{array} \right| \end{array} = 10$$

$$W_{1011} = \begin{array}{c} \left| \begin{array}{c} 0111 \\ 1011 \end{array} \right| + \left| \begin{array}{c} 0011 \\ 1011 \end{array} \right| + \left| \begin{array}{c} 1011 \\ 0010 \end{array} \right| + \left| \begin{array}{c} 1011 \\ 1011 \end{array} \right| + \left| \begin{array}{c} 1011 \\ 0000 \end{array} \right| \end{array} = 8$$

$$W_{1010} = \begin{array}{c} \left| \begin{array}{c} 0111 \\ 1010 \end{array} \right| + \left| \begin{array}{c} 0011 \\ 1010 \end{array} \right| + \left| \begin{array}{c} 1010 \\ 0010 \end{array} \right| + \left| \begin{array}{c} 1010 \\ 1010 \end{array} \right| + \left| \begin{array}{c} 1010 \\ 0000 \end{array} \right| \end{array} = 8$$

$$W_{1000} = \begin{array}{c} \left| \begin{array}{c} 0111 \\ 1000 \end{array} \right| + \left| \begin{array}{c} 0011 \\ 1000 \end{array} \right| + \left| \begin{array}{c} 1000 \\ 0010 \end{array} \right| + \left| \begin{array}{c} 1000 \\ 1000 \end{array} \right| + \left| \begin{array}{c} 1000 \\ 0000 \end{array} \right| \end{array} = 10$$

Обираємо код з мінімальним значенням W .

$K_9 = 1010$.

Завершення розрахунків.

Приклад логічного парадоксу при вирішенні задачі щодо двоканальної СМО з необмеженою чергою

На двох комп'ютерах у Центрі підготовки персоналу деякої компанії XYQ встановлені програми для тестування знань з двох напрямків роботи компанії. Персонал, що приходить на тестування, шикується в одну чергу – як звільняється якийсь із комп'ютерів, найближчий в черзі його займає. Інтенсивність потоку респондентів для обох напрямків роботи однакове: $\lambda_A = \lambda_B = 0,45$ (респондента на хвилину), а в сумі вони утворюють загальний потік з інтенсивністю $\lambda_A + \lambda_B = 0,9$.

В середньому тестування займає 2 хвилини. Але респонденти почали скаржитися, що вистоювання в черзі відволікає їх від роботи.

Надійшла раціоналізаторська пропозиція: замість одного комп'ютера, що одночасно містить тести по двом напрямкам роботи, створити два спеціалізованих, кожен з яких утримуватиме лише один тест за напрямком. Розумність цієї пропозиції викликає суперечки – дехто стверджує, що черги залишаться колишніми. Вимагається перевірити корисність пропозиції розрахунком.

Припускаємо, що усі потоки подій – прості (на якісній стороні висновків це не позначиться). Розглянемо два варіанти організації роботи комп'ютерів – існуючий і пропонований.

Варіант I (існуючий). На двоканальну СМО поступає потік респондентів з інтенсивністю $\lambda = 0,9$; інтенсивність потоку обробки тестів $\mu = 1/2 = 0,5$; $\rho = \lambda/\mu = 1,8$. Оскільки $\rho > 1$, фінальна ймовірність існує. Результат $p_0 \approx 0,0525$. Середнє число оброблених результатів в черзі $L_{\text{чер}} = 7,68$; середній час, що проводиться в черзі $W_{\text{чер}} = 8,54$ (хв.).

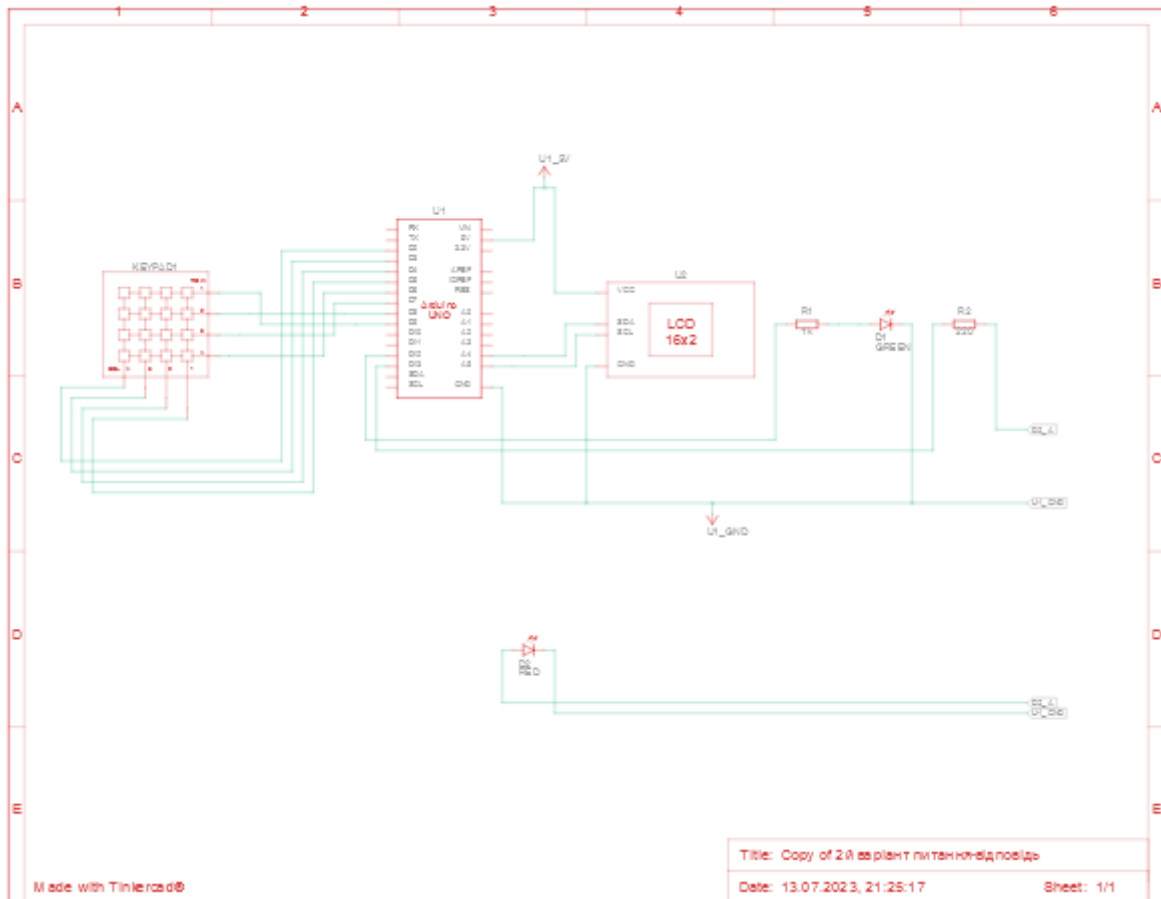
Варіант II (пропонований). Потрібно розглянути дві одноканальні СМО (два спеціалізованих комп'ютера). На кожен поступає потік респондентів з

інтенсивністю $\lambda = 0,45$; μ як і раніше рівне $0,5$; $\rho = \lambda/\mu = 0,9 < 1$; фінальна ймовірність існує. Середня довжина черги $L_{\text{чер}} = 8,1$.

Спостережуваний парадокс. Довжина черги, виявляється, не лише не зменшилася, а збільшилася! Ділячи $L_{\text{чер}}$ на $\lambda = 0,45$, отримаємо середній час очікування $W_{\text{чер}} = 18$ (хвилин). Тобто, замість того щоб зменшитися, середня довжина черги і середній час очікування в ній збільшилися.

Реалізація комп'ютерного експерименту з використання абстрактного автомату для створення системи імітації людської поведінки в процесі навчання

Г 1. Семантична схема реалізації



Г 2. Використані компоненти Arduino

| Name | Quantity | Component |
|---------|----------|------------------------------------|
| U1 | 1 | Arduino Uno R3 |
| KEYPAD1 | 1 | Keypad 4x4 |
| U2 | 1 | PCF8574-based, 39 LCD 16 x 2 (I2C) |
| D1 | 1 | Green LED |
| R1 | 1 | 1 k Ω Resistor |
| D2 | 1 | Red LED |
| R2 | 1 | 220 Ω Resistor |

Г 3. Лістинг коду

```

#include <Wire.h>
#include <LiquidCrystal_I2C.h>
#include <Keypad.h>

// Підключення LCD
LiquidCrystal_I2C lcd(0x27, 16, 2); // Адреса I2C дисплея та
його розмірність (16 символів на 2 рядка)

// Підключення клавіатури
const byte ROWS = 4; // кількість рядків
const byte COLS = 4; // кількість стовпців
char keys[ROWS][COLS] = {
  {'1', '2', '3', 'A'},
  {'4', '5', '6', 'B'},
  {'7', '8', '9', 'C'},
  {'*', '0', '#', 'D'}
};
byte rowPins[ROWS] = {9, 8, 7, 6}; // піни, під'єднані до
рядків клавіатури
byte colPins[COLS] = {5, 4, 3, 2}; // // піни, під'єднані до
стовпців клавіатури
Keypad keypad = Keypad(makeKeypad(keys), rowPins, colPins, ROWS,
COLS);

// Піни світлодіодів
const int greenLedPin = 12;
const int redLedPin = 13;

// Питання
const char question[] = "How many fingers are on a human hand?
";
// Правильна відповідь
const int expectedAnswer = 5;

// Швидкість прокручування тексту (у мілісекундах)
const int scrollSpeed = 1;

void setup() {
  lcd.init(); // Ініціалізація LCD
  lcd.backlight(); // Включення підсвітки LCD

  // налаштування пінів зі світлодіодами як вихідних
  pinMode(greenLedPin, OUTPUT);
  pinMode(redLedPin, OUTPUT);
}

void loop() {

```

```
// Прокручування тексту на верхньому ряду
scrollText(question, 0, 0, scrollSpeed);

// Виведення повідомлення для відповіді на нижньому рядку
lcd.setCursor(0, 1);
lcd.print("Enter answer: ");

// Очікування введення відповіді користувача
char key = NO_KEY;
while (key == NO_KEY) {
    key = keypad.getKey();
}

// Перевірка відповіді
int answer = key - '0'; // Перетворення символу на число

// Очищення екрану
lcd.clear();

if (answer == expectedAnswer) {
    // Правильна відповідь, включаємо зелений світлодіод і
    виводимо повідомлення на LCD
    digitalWrite(greenLedPin, HIGH);
    digitalWrite(redLedPin, LOW);
    lcd.print("Correct!");
} else {

    // Неправильна відповідь, включаємо червоний світлодіод та
    виводимо повідомлення на LCD

    digitalWrite(greenLedPin, LOW);
    digitalWrite(redLedPin, HIGH);
    lcd.print("Wrong!");
}

// Зачекати деякий час перед наступним питанням
delay(2000);

// Очистити екран та скинути світлодіоди
lcd.clear();
digitalWrite(greenLedPin, LOW);
digitalWrite(redLedPin, LOW);
}

// Функція для прокручування тексту на LCD
void scrollText(const char* text, int col, int row, int speed) {
    int textLength = strlen(text);
```

```
for (int i = 0; i <= textLength - 16; i++) {  
    lcd.setCursor(col, row);  
    lcd.print(text + i);  
    delay(speed);  
    lcd.clear();  
}  
}
```

Лістинг коду реалізації 3D резюме

Д 1. Код головної сторінки

```

<template>
  <div class="home">
    <BaseHeader />
    <div class="home__wrap">
      <div class="home__title">Етапи тестування</div>
      <div class="home__tests">
        <router-link :to="test.link" class="test"
: class="test.id" v-for="test in cards" :key="test.id">
          <div class="test__top">
            <div class="test__icon" :class="test.id"></div>
            <div class="test__title">{{ test.title }}</div>
            <div class="test__info">
              <div class="test__info__line" v-for="(line, index)
in test.infoLines" :key="index">
                <div class="test__info__line__name">{{
line.title }}</div>
                <div class="test__info__line__value">{{
line.value }}</div>
              </div>
            </div>
          </div>
          <div class="test__bottom">
            <div class="test__btn">розпочати</div>
            <div class="test__btn-ico"></div>
          </div>
          <div class="test__hover">
            <svg xmlns="http://www.w3.org/2000/svg" width="356"
height="599" viewBox="0 0 356 599" fill="none">
              <path fill-rule="evenodd" clip-rule="evenodd"
d="M30 0C13.4315 0 0 13.4315 0 30V449C0 465.569 13.4315 479 30
479C13.4315 479 0 492.431 0 509V569C0 585.569 13.4315 599 30
599H326C342.569 599 356 585.569 356 569V509C356 492.431 342.569
479 326 479C342.569 479 356 465.569 356 449V30C356 13.4315
342.569 0 326 0H30Z" fill="url(#paint0_radial_45_420)"/>
              <defs>
                <radialGradient id="paint0_radial_45_420" cx="0"
cy="0" r="1" gradientUnits="userSpaceOnUse"
gradientTransform="translate(78.6947 190.079) rotate(76.3501)
scale(420.806 260.983)">
                  <stop stop-color="#2C16B4"/>
                  <stop offset="1" stop-color="#15181D"/>
                </radialGradient>
              </defs>
            </svg>
          </div>
        </router-link>

```

```

        </div>
    </div>
</div>
</template>

<script>
// @ is an alias to /src
import BaseHeader from "@components/BaseHeader.vue";

export default {
  name: 'HomeView',
  components: {
    BaseHeader,
  },
  data() {
    return {
      cards: [
        {
          id: 'testing',
          link: '/testing',
          title: 'Тестування',
          infoLines: [
            {
              title: 'Час проходження',
              value: '40 хв.',
            },
            {
              title: 'Кількість спроб',
              value: '1',
            },
          ],
        },
        {
          id: 'task',
          link: '/task',
          title: 'Тестове завдання',
          infoLines: [
            {
              title: 'Час проходження',
              value: '20 хв.',
            },
            {
              title: 'Кількість спроб',
              value: '1',
            },
          ],
        },
        {
          id: 'video',
          link: '/video',
          title: 'Відеопитання',
          infoLines: [
            {

```

```

        title: 'Час проходження',
        value: '20 хв.',
      },
      {
        title: 'Кількість спроб',
        value: '1',
      },
    ],
  },
]
}
}
</script>

```

```

<style scoped lang="scss">
  .home {
    position: relative;

    &__wrap {
      margin: 0 40px 40px;
      background: #fff;
      border-radius: 50px;
      padding: 60px 50px 50px;
    }

    &__title {
      color: #15181D;
      text-align: center;
      font-family: Neue Machina, serif;
      font-size: 40px;
      font-style: normal;
      font-weight: 900;
      line-height: normal;
      margin: 0 0 40px;
    }

    &__tests {
      display: flex;
      justify-content: center;
      align-items: center;
    }

    .test {
      color: #fff;
      width: 356px;
      position: relative;
      overflow: hidden;
      border-radius: 30px;
      cursor: pointer;
      text-decoration: none;

      &__hover {

```

```

    position: absolute;
    left: 0;
    bottom: 0;
    display: flex;
    opacity: 0;
    transition-duration: .3s;
}
&__top {
    border-radius: 25px;
    background: #15181d;
    position: relative;
    padding: 50px 40px;

    > div {
        position: relative;
        z-index: 2;
    }
}
&__bottom {
    border-radius: 25px;
    background: #15181d;
    padding: 30px 40px;
    display: flex;
    align-items: center;

    > div {
        position: relative;
        z-index: 2;
    }
}

&__icon {
    height: 182px;
    width: 100%;
    margin: 20px 0;

    &.testing {
        background: url(../assets/images/ico_q_1.png) no-
repeat 0 center;
    }
    &.task {
        background: url(../assets/images/ico_q_2.png) no-
repeat 0 center;
    }
    &.video {
        background: url(../assets/images/ico_q_3.png) no-
repeat 0 center;
    }
}

&__title {
    color: #FFF;
    font-family: Neue Machina, serif;
}

```

```

    font-size: 32px;
    font-style: normal;
    font-weight: 900;
    line-height: normal;
    height: 32px;
    display: flex;
    align-items: flex-end;
    margin-bottom: 25px;
}

&__info {
  &__line {
    display: flex;
    align-items: center;
    justify-content: space-between;
    margin-bottom: 5px;
  }
}

&__btn {
  flex: 1 1 100%;
  padding-top: 2px;
  border: 1px solid #fff;
  border-radius: 50px;
  text-align: center;
  font-family: Neue Machina, serif;
  font-size: 20px;
  font-style: normal;
  font-weight: 700;
  line-height: 54px; /* 150% */
  letter-spacing: 1px;
  text-transform: uppercase;
  cursor: pointer;

  &-ico {
    border-radius: 50px;
    border: 1px solid #2C16B4;
    background: #2C16B4 url(../assets/images/arrow-w.svg)
no-repeat 90% 50%;
    width: 58px;
    height: 58px;
    min-width: 58px;
    cursor: pointer;
  }
}

&:hover {
  .test {
    &__hover {
      opacity: 1;
    }
    &__btn {
      background: #2C16B4;
    }
  }
}

```



```

        <span v-if="test.value">Відповідь
зарахована</span>
        <span v-else-if="test.id === currentTest.id">В
процесі</span>
        <span v-else>Очікує відповіді</span>
    </div>
</div>
    <div class="testing__list__item__status"
:class="{ 'has-answer': test.value, current: test.id ===
currentTest.id }"></div>
    </div>
</div>
</div>
</div>
</template>

```

```

<script>
// @ is an alias to /src
import BaseHeader from "@components/BaseHeader.vue";
import BaseSuccess from "@components/BaseSuccess.vue";

export default {
  name: 'TaskView',
  components: {
    BaseSuccess,
    BaseHeader,
  },
  data() {
    return {
      currentTest: null,
      currentTestNumber: null,
      error: '',
      answers: [
        {
          id: 1,
          name: 'Абсолютно невірно'
        },
        {
          id: 2,
          name: 'Скоріш невірно'
        },
        {
          id: 3,
          name: 'Скоріш вірно'
        },
        {
          id: 4,
          name: 'Абсолютно вірно'
        },
      ],
      testingList: [
        {
          id: 1,

```

```
    number: '№1',
    question: 'Питання',
    value: null,
  },
  {
    id: 2,
    number: '№2',
    question: 'Питання',
    value: null,
  },
  {
    id: 3,
    number: '№3',
    question: 'Питання',
    value: null,
  },
  {
    id: 4,
    number: '№4',
    question: 'Питання',
    value: null,
  },
  {
    id: 5,
    number: '№5',
    question: 'Питання',
    value: null,
  },
  {
    id: 6,
    number: '№6',
    question: 'Питання',
    value: null,
  },
  {
    id: 7,
    number: '№7',
    question: 'Питання',
    value: null,
  },
  {
    id: 8,
    number: '№8',
    question: 'Питання',
    value: null,
  },
  {
    id: 9,
    number: '№9',
    question: 'Питання',
    value: null,
  },
  {
```

```

        id: 10,
        number: '№10',
        question: 'Питання',
        value: null,
      },
    ],
  },
  computed: {
    hasSuccessProcess() {
      return this.$store.state.testingProcess;
    }
  },
  methods: {
    saveAnswer() {
      if (this.currentTest.value) {
        this.testingList[this.currentTestNumber].value =
this.currentTest.value;

        if (this.currentTestNumber < (this.testingList.length -
1)) {
          this.currentTestNumber++;
          this.currentTest = Object.assign({},
this.testingList[this.currentTestNumber]);
        } else {
          this.$store.commit('setItem', {item: 'testingProcess',
data: true})
          this.$store.commit('setItem', {item: 'testingValue',
data: this.testingList})
        }
      } else {
        this.error = "Оберіть варіант відповіді"
      }
    }
  },
  mounted() {
    this.currentTest = Object.assign({}, this.testingList[0]);
    this.currentTestNumber = 0;
  }
}
</script>

<style scoped lang="scss">
.testing {
  position: relative;

  &__wrap {
    margin: 0 40px 40px;
    background: #fff;
    border-radius: 50px;
    padding: 100px 100px;
    display: flex;
    align-items: flex-start;
  }
}

```

```

}

&__title {
  margin-bottom: 40px;
  font-family: Neue Machina, serif;
  font-size: 40px;
  font-style: normal;
  font-weight: 900;
  line-height: 60px;
}

&__current {
  flex: 1 1 100%;
}

&__list {
  width: 300px;
  min-width: 300px;
  padding-left: 40px;

  &__item {
    padding: 10px 20px;
    border: 1px solid #2C16B4;
    margin-bottom: -1px;
    border-radius: 20px;
    display: flex;
    align-items: center;
    justify-content: space-between;
    background: #D7E2EC;
    font-size: 14px;
    font-weight: 500;
    line-height: 1;
    width: 100%;

    &.current,
    &.has-answer {
      background: transparent;
    }

    &__name {
      font-family: Neue Machina, serif;
      font-size: 20px;
      font-style: normal;
      font-weight: 700;
    }

    &__status {
      min-width: 30px;
      width: 30px;
      height: 30px;
      border: 2px solid #fff;
      border-radius: 50%;
    }
  }
}

```

```

        &.has-answer {
            background: url(../assets/images/chack.svg) no-
repeat center;
            background-size: cover;
            border-color: #3CE494;
        }
        &.current {
            background: transparent;
            border: 2px dashed #2C16B4;
        }
    }
}

.test {
    &__name {
        color: #000;
        font-family: Neue Machina, serif;
        font-size: 24px;
        font-weight: 700;
        line-height: normal;
        margin-bottom: 20px;
    }

    &__question {
        font-size: 20px;
        font-weight: 500;
        line-height: 35px;
        margin-bottom: 40px;
    }

    &__answers {
        position: relative;

        &__error {
            color: #EC2E5C;
        }

        &__item {
            margin-bottom: 20px;

            input[type=radio] {
                display: none;

                & + label {
                    display: flex;
                    align-items: center;
                    justify-content: flex-start;
                    position: relative;
                    cursor: pointer;
                    font-family: Neue Machina, serif;
                    font-size: 20px;
                    font-style: normal;
                }
            }
        }
    }
}

```

```

font-weight: 700;

&::before {
  content: '';
  display: block;
  width: 30px;
  min-width: 30px;
  height: 30px;
  border: 2px dashed #2C16B4;
  border-radius: 50%;
  margin-right: 15px;
}
}

&:checked + label {
  &:before {
    border: 2px solid #2C16B4;
  }
  &:after {
    content: '';
    width: 22px;
    height: 22px;
    background: #2C16B4;
    border-radius: 50%;
    position: absolute;
    left: 6px;
    top: calc(50% - 11px);
  }
}
}
}
}

&__btn {
  display: inline-flex;
  align-items: center;
  cursor: pointer;
  margin-top: 40px;

  &--text {
    width: 220px;
    background: #2C16B4;
    line-height: 58px;
    border-radius: 29px;
    color: #fff;
    text-align: center;
    font-family: Neue Machina, serif;
    font-size: 20px;
    font-style: normal;
    font-weight: 700;
    letter-spacing: 1px;
    text-transform: uppercase;
  }
}

```

```

        &--ico {
            background: url(../assets/images/arrow-d.svg) no-
repeat 90% 50%;
            border: 2px solid #2C16B4;
            border-radius: 50%;
            width: 58px;
            height: 58px;
        }
    }
}
}
}
</style>

```

Д 3. Код сторінки тестового питання

```

<template>
  <div class="task">
    <BaseHeader />
    <base-success v-if="hasSuccessProcess" />
    <div class="task__wrap" v-else>
      <div class="task__title">Тестове завдання</div>
      <div class="task__description">Опишіть ... (суть
питання).</div>
      <div class="task__editor">
        <mc-wysiwyg v-model="html" :hide="hide" :height="200"
@input="error=''"></mc-wysiwyg>
      </div>
      <div class="task__error" v-if="error">{{ error }}</div>
      <div class="task__btn" @click="saveAnswer">
        <div class="task__btn--text">підтвердити</div>
        <div class="task__btn--ico"></div>
      </div>
    </div>
  </div>
</template>

```

```

<script>
// @ is an alias to /src
import BaseHeader from "@/components/BaseHeader.vue";
import { McWysiwyg } from '@mucure/vue-wysiwyg';
import BaseSuccess from "@/components/BaseSuccess.vue";
export default {
  name: 'TaskView',
  components: {
    BaseSuccess,
    BaseHeader,
    McWysiwyg
  },
  data() {
    return {
      error: '',
      html: '',

```



```

    hide: {
      strikethrough: true,
      table: true,
      heading: true,
      url: true,
      indent: true,
      outdent: true,
    }
  },
  computed: {
    hasValue() {
      let value = this.html.replace(/&nbsp;/gi, '');
      return value.trim().length;
    },
    hasSuccessProcess() {
      return this.$store.state.taskProcess;
    }
  },
  methods: {
    saveAnswer() {
      if (this.hasValue) {
        this.$store.commit('setItem', {item: 'taskProcess',
data: true})
        this.$store.commit('setItem', {item: 'taskValue', data:
this.html})
      } else {
        this.error = 'Введіть відповідь';
      }
    }
  }
}
</script>

<style scoped lang="scss">
.task {
  position: relative;

  &__wrap {
    margin: 0 40px 40px;
    background: #fff;
    border-radius: 50px;
    padding: 80px 100px;
  }

  &__title {
    color: #15181D;
    font-family: Neue Machina, serif;
    font-size: 40px;
    font-style: normal;
    font-weight: 900;
    line-height: 70px;
    margin-bottom: 40px;
  }
}

```

```

}

&__description {
  width: 1120px;
  font-size: 20px;
  font-style: normal;
  font-weight: 500;
  line-height: 35px;
  margin-bottom: 40px;
}

&__error {
  color: #EC2E5C;
  padding: 0 20px;
}

&__btn {
  display: inline-flex;
  align-items: center;
  cursor: pointer;
  margin-top: 40px;

  &--text {
    width: 220px;
    background: #2C16B4;
    line-height: 58px;
    border-radius: 29px;
    color: #fff;
    text-align: center;
    font-family: Neue Machina, serif;
    font-size: 20px;
    font-style: normal;
    font-weight: 700;
    letter-spacing: 1px;
    text-transform: uppercase;
  }

  &--ico {
    background: url(../assets/images/arrow-d.svg) no-repeat
90% 50%;
    border: 2px solid #2C16B4;
    border-radius: 50%;
    width: 58px;
    height: 58px;
  }
}
}
</style>

<style lang="scss">
.task__editor {
  #container {
    border: 0 !important;

```

```

}
#toolbar {
  background: #F6F9FC !important;
  border-radius: 20px;
  border: 0 !important;
  padding: 10px 40px;
  height: 50px;
  display: flex;
  align-items: center;

  .toolbar-section {
    margin-right: 30px;
    border: 0 !important;
    display: flex;
    align-items: center;

    .wysiwyg-button {
      padding: 0;
      width: 30px;
      height: 30px;
      border: 0 solid #2C16B4;
      color: #2C16B4;
      margin-right: 10px;
      display: flex;
      align-items: center;
      justify-content: center;
      border-radius: 10px;

      &:hover {
        background: #2C16B4;
        color: #fff;
      }

      .material-design-icon {
        display: flex;
        align-items: center;
        justify-content: center;
        width: 28px;
        height: 28px;
      }
      svg {
        width: 24px;
        height: 24px;
        margin-bottom: 1px;
        position: relative;
      }
    }
  }
}
#editor {
  padding: 30px 40px !important;
  font-size: 20px;
  line-height: 35px;

```

```

    color: #15181D;
  }
  #body {
    background: #F6F9FC !important;
    border-radius: 20px;

    &::-webkit-scrollbar {
      width: 6px;
    }
    &::-webkit-scrollbar-thumb {
      background: #2C16B4;
      border-radius: 3px;
    }
  }
}
</style>

```

Д 4. Код сторінки відео питання

```

<template>
  <div class="video">
    <BaseHeader />
    <base-success v-if="hasSuccessProcess" />
    <div class="video__wrap" v-else>
      <div class="video__block" v-if="!showFields">
        <div class="video__title" v-
show="!isPlaying">Відеопитання</div>
        <div class="video__content" :class="{playing:
isplaying}" @click="playVideo">
          <video :src="question" :poster="poster" ref="video"
@ended="endPlay" />
          <div class="video__content--back" v-
if="!isPlaying"></div>
        </div>
        <div class="video__btn dark" v-show="!isPlaying"
@click="showFields = true">
          <div class="video__btn--text">Відповісти</div>
          <div class="video__btn--ico"></div>
        </div>
      </div>
      <div class="video__answer" v-else>
        <div class="video__head">
          <div class="video__title">Відеопитання</div>
          <div class="video__review" @click="showFields =
false">Переглянути ще раз</div>
        </div>
        <div class="video__editor">
          <mc-wysiwyg v-model="html" :hide="hide" :height="200"
@input="error = ''"></mc-wysiwyg>
        </div>
        <div class="video__error" v-if="error">{{ error }}</div>
        <div class="video__btn" @click="saveAnswer">

```

```

        <div class="video__btn--text">підтвердити</div>
        <div class="video__btn--ico"></div>
    </div>
</div>
</div>
</template>

```

```

<script>
// @ is an alias to /src
import BaseHeader from "@components/BaseHeader.vue";
import question from '../assets/question.mp4';
import poster from '../assets/poster.webp';
import { McWysiwyg } from '@mycure/vue-wysiwyg';
import BaseSuccess from "@components/BaseSuccess.vue";

export default {
  name: 'VideoView',
  components: {
    BaseSuccess,
    BaseHeader,
    McWysiwyg
  },
  data() {
    return {
      question: question,
      poster: poster,
      isPlaying: false,
      showFields: false,
      error: '',
      html: '',
      hide: {
        strikethrough: true,
        table: true,
        heading: true,
        url: true,
        indent: true,
        outdent: true,
      }
    }
  },
  computed: {
    hasValue() {
      let value = this.html.replace(/&nbsp;/gi, '');
      return value.trim().length;
    },
    hasSuccessProcess() {
      return this.$store.state.videoProcess;
    }
  },
  methods: {
    playVideo() {
      if (this.isPlaying) {

```

```

        this.isPlaying = false;
        this.$refs.video.pause()
    } else {
        this.isPlaying = true;
        this.$refs.video.play()
    }
},
endPlay() {
    this.isPlaying = false;
    this.$refs.video.currentTime = 0;
},
saveAnswer() {
    if (this.hasValue) {
        this.$store.commit('setItem', {item: 'videoProcess',
data: true})
        this.$store.commit('setItem', {item: 'videoValue', data:
this.html})
    } else {
        this.error = 'Введіть відповідь';
    }
}
}
}
</script>

```

```

<style scoped lang="scss">
    .video {
        position: relative;

        &__wrap {
            margin: 0 40px 40px;
        }

        &__title {
            font-family: Neue Machina, serif;
            font-size: 40px;
            font-style: normal;
            font-weight: 900;
            line-height: 70px;
            margin-bottom: 40px;
        }

        &__head {
            display: flex;
            align-items: flex-start;
        }

        &__review {
            border: 2px solid #2C16B4;
            color: #2C16B4;
            padding: 0 20px;
            line-height: 54px;
            font-family: Neue Machina, serif;
        }
    }

```

```

font-size: 20px;
font-style: normal;
font-weight: 700;
letter-spacing: 1px;
text-transform: uppercase;
border-radius: 29px;
margin-left: 30px;
margin-top: 5px;
cursor: pointer;
}

&__btn {
display: inline-flex;
align-items: center;
cursor: pointer;
margin-top: 40px;

  &--text {
width: 220px;
background: #2C16B4;
line-height: 58px;
border-radius: 29px;
color: #fff;
text-align: center;
font-family: Neue Machina, serif;
font-size: 20px;
font-style: normal;
font-weight: 700;
letter-spacing: 1px;
text-transform: uppercase;
}

  &--ico {
background: url(../assets/images/arrow-d.svg) no-repeat
90% 50%;
border: 2px solid #2C16B4;
border-radius: 50%;
width: 58px;
height: 58px;
}

  &.dark {
  .video__btn--ico {
background: url(../assets/images/arrow-w.svg) no-
repeat 90% 50%;
}
}
}

&__block {
color: #fff;
position: relative;

```

```

height: calc((100vw - 40px) * 0.54);
max-height: calc(100vh - 160px);
border-radius: 50px;
overflow: hidden;

.video {
  &__title {
    position: absolute;
    z-index: 2;
    left: 100px;
    top: 100px;
  }

  &__content {
    width: 100%;
    height: 100%;
    position: absolute;
    left: 0;
    top: 0;
    z-index: 1;
    cursor: pointer;

    video {
      width: 100%;
      height: 100%;
      background: #fff;
      outline: none !important;
    }

    &--back {
      width: 100%;
      height: 100%;
      position: absolute;
      left: 0;
      top: 0;
      background: rgba(21, 24, 29, 0.8)
url(../assets/play-icon.png) no-repeat center;
    }
  }

  &__btn {
    position: absolute;
    bottom: 100px;
    left: calc(50% - 139px);
    z-index: 2;
  }
}

&__answer {
  background: #fff;
  border-radius: 50px;
  padding: 60px 50px 50px;
}

```



```

        .video__title {
            }
        }

        &__error {
            color: #EC2E5C;
            padding: 0 20px;
        }
    }
</style>

<style lang="scss">
    .video__editor {
        #container {
            border: 0 !important;
        }
        #toolbar {
            background: #F6F9FC !important;
            border-radius: 20px;
            border: 0 !important;
            padding: 10px 40px;
            height: 50px;
            display: flex;
            align-items: center;

            .toolbar-section {
                margin-right: 30px;
                border: 0 !important;
                display: flex;
                align-items: center;

                .wysiwyg-button {
                    padding: 0;
                    width: 30px;
                    height: 30px;
                    border: 0 solid #2C16B4;
                    color: #2C16B4;
                    margin-right: 10px;
                    display: flex;
                    align-items: center;
                    justify-content: center;
                    border-radius: 10px;

                    &:hover {
                        background: #2C16B4;
                        color: #fff;
                    }
                }

                .material-design-icon {
                    display: flex;
                    align-items: center;
                }
            }
        }
    }

```

```
        justify-content: center;
        width: 28px;
        height: 28px;
    }
    svg {
        width: 24px;
        height: 24px;
        margin-bottom: 1px;
        position: relative;
    }
}
}
}
#editor {
    padding: 30px 40px !important;
    font-size: 20px;
    line-height: 35px;
    color: #15181D;
}
#body {
    background: #F6F9FC !important;
    border-radius: 20px;

    &::-webkit-scrollbar {
        width: 6px;
    }
    &::-webkit-scrollbar-thumb {
        background: #2C16B4;
        border-radius: 3px;
    }
}
}
</style>
```

**Приклад реалізації алгоритму виконання процедури обчислення
відповідності ключовому слову**

На вході: приймаємо $N = 100, q = 32, n = 4, m = 10, r = \frac{N}{q^n} \approx 10^{-3}$.

Чисельне інтерпретування алгоритму:

$$A = 00; \quad B = 01; \quad B = 02; \quad \dots$$

Гіпотетичне ключове слово – ШИФР.

$$a_1 = Б; \hat{a}_1 = Б; A_0^H(\hat{a}_1) = 01.00.00.00 = 1 \cdot 32^3 = 32768;$$

$$AD_1^H(\hat{B}) = 32768 \cdot 10^{-3} \approx 32;$$

$$\Delta_1 = 32768 - 33792 = -1024 \neq 0.$$

$$AD_2^H(\hat{B}) \approx 32 - 1 = 31;$$

$$A(31) = СТАН = 33760;$$

$$\Delta_2 = 32768 - 33760 \neq 0.$$

.....

$$AD^H(\hat{B}) = 30;$$

.....

$$AD^K(\hat{B}) = 63;$$

$$m(\hat{B}) = 63 - 30 + 1 = 34 > 10.$$

Процес вводу символів триває;

$$\text{т.к. } m(\hat{B}) > m.$$

$$a_2 = Я; \hat{a}_2 = (БЯ);$$

$$A_0^H(\hat{a}_2) = 01.31.00.00 = 64512;$$

$$AD_1^H(\hat{a}_2) = 64512 \cdot 10^{-3} \approx 64;$$

$$A(64) = КЛЮЧ = 2 \cdot 32^3 = 65536;$$

$$\Delta_1 = -1024 \neq 0.$$

$$AD_2^H(\hat{a}_2) = 63;$$

$$A(63) = \text{КОДИ} = 65472;$$

.....

$$AD^H(\hat{a}_2) = 61;$$

.....

$$AD^K(\hat{a}_2) = 63;$$

$$m(\hat{a}_2) = 3 < 10.$$

Процес вводу символів завершено, $v_d = 2$.

Відповідь користувача приймається, бо зазначені ключові слова, що є у базовому словнику системи: СТАН, КЛЮЧ, КОДИ.

Данні математичного і імітаційного моделювання

| Сигнатура | $\bar{v}_1^{(2)}$ | | | $\bar{T}_{(мс)}$ | $\bar{v}_1^{(3)}$ | | | $\bar{T}_{(мс)}$ |
|--|---------------------------------|--------------------------------|-----------------|------------------|---------------------------------|--------------------------------|-----------------|------------------|
| | Розрахунок $\bar{v}_1^{(2)}$ | Експеримент $\bar{V}^{(2)}$ | Розрах./експер. | | Розрахунок $\bar{v}_1^{(3)}$ | Експеримент $\bar{V}^{(3)}$ | Розрах./експер. | |
| 1. ТЕКСТ $N = 1,1 \cdot 10^6, r = 10^{-6}$ | 3,999 | 4,005 | ~1,0 | 13,3 | 4,67 | 4,68 | 0,998 | 13,4 |
| 2. ТЕКСТ $N = 1,1 \cdot 10^4, r = 10^{-8}$ | 2,63 | 2,65 | 0,992 | ≤ 1 | 3,29 | 3,30 | 0,996 | 1,25 |
| 3. ТЕКСТ $N = 1,1 \cdot 10^3, r = 10^{-9}$ | 1,999 | 2,00 | ~1,0 | ≤ 1 | 2,68 | 2,70 | 0,992 | ≤ 1 |
| 4. ТЕКСТ $N = 1,1 \cdot 10^2, r = 10^{-10}$ | 1,003 | 1,00 | ~1,0 | ≤ 1 | 2,07 | 2,08 | 0,995 | ≤ 1 |
| 5. КОД $N = 1 \cdot 10^6, r = 10^{-6}$ | 5,54 | 5,55 | 0,998 | 39,7 | 6,69 | 6,76 | 0,989 | 40,6 |
| 6. КОД $N = 1 \cdot 10^4, r = 10^{-8}$ | 3,54 | 3,55 | 0,997 | ≤ 1 | 4,69 | 4,74 | 0,989 | 1,2 |
| 7. КОД $N = 1 \cdot 10^3, r = 10^{-9}$ | 2,57 | 2,58 | 0,98 | ≤ 1 | 3,69 | 3,72 | 0,99 | ≤ 1 |
| 8. КОД $N = 1 \cdot 10^2, r = 10^{-10}$ | 1,54 | 1,60 | 0,96 | ≤ 1 | 2,69 | 2,73 | 0,985 | ≤ 1 |