



МІНІМАЛЬНЕ ПОДАННЯ СКІНЧЕННИХ ПРЕДИКАТИВ ДОВІЛЬНОГО ПОРЯДКУ ПРИ ПОБУДОВІ АДАПТИВНИХ СИСТЕМ

УДК 519.7:681.3

ГОРБАЧ Татьяна Викторовна

научный сотрудник кафедры Программной инженерии Харьковского национального университета радиоэлектроники

Научные интересы: Разработка и внедрение компьютерных обучающих программ, разработка алгоритмического и программного обеспечения для дистанционного образования, разработка программных средств мультимедиа.

e-mail: yova.tanya@gmail.com

ГОНЧАРОВ Петр Владиславович

аспирант кафедры Программной инженерии Харьковского национального университета радиоэлектроники

Научные интересы: Разработка и внедрение компьютерных обучающих программ, разработка алгоритмического и программного обеспечения для дистанционного образования, разработка программных средств мультимедиа

e-mail: pitergon@list.ru

СНИСАР Станислав Михайлови

аспирант кафедры Программной инженерии Харьковского национального университета радиоэлектроники,

Научные интересы: Математическое моделирование, алгебро-логические структуры, развитие теории и практических применений структур алгебры конечных предикатов.

e-mail: sn.stanislav4@gmail.com

ШУБИН Игорь Юрьевич

кандидат технических наук, доцент, профессор кафедры Программной инженерии

Харьковского национального университета радиоэлектроники

Научные интересы: Математическое моделирование, разработка и внедрение компьютерных обучающих программ, разработка алгоритмического и программного обеспечения для дистанционного образования, разработка программных средств мультимедиа

e-mail: igor.shubin@nure.ua

ВСТУП

Проведений аналіз науково-практичних розробок з математичного забезпечення моделей адаптації в сучасних інтелектуальних гіпермедіа системах доводить їх придатність для опису умов і вимог дистанційного навчання. Для досягнення мети адаптації система повинна мати інформацію для аналізу інтересів та вполюбовань користувача, історії його взаємодії з системою, будь-яку інформацію, до якої вона може адаптуватися. Окремим питанням постають види та варіанти надання інформації, вирішення на конкретному етапі питання релевантності наданої інформації, та інше. За своєю структурою адаптивні системи дуже різноманітні, від

надзвичайно складних до більш простих, з меншою кількістю компонентів та параметрів, що враховуються при побудові сценарію навчання. Основними компонентами адаптивних навчальних систем є моделі користувача та предметної галузі, а також, в залежності від типу системи, база знань чи інші математичні моделі [1]. Робота є одним із напрямків наукових досліджень кафедри Програмної інженерії ХНУРЕ, дослідження перш за все пов'язані з системами дистанційного навчання та впровадженням мультимедійних технологій у процес навчання [2].

ПОСТАНОВКА ЗАДАЧІ

Одним із напрямків сучасних досліджень та розробки математичних моделей представлення знань і обробки неоднорідної (гіпермедійної) інформації є моделі адаптації в інтелектуальних гіпермедіасистемах для дистанційного навчання. Також необхідно провести їх порівняльний аналіз щодо використання сучасних методів моделювання процесів навчання з метою підвищення ефективності отримання знань. Також аналіз стану вирішення проблеми вказує на необхідність розвитку математичного апарату моделювання інтелектуальних функцій людини для формалізації процесу навчання та адаптації навчальних матеріалів при побудові інтелектуальних інформаційних систем.

На підставі результатів аналізу наявних проблем і завдань у галузі автоматизації розробки інтелектуальних адаптивних навчальних гіпермедіасистем сформульовано та обґрунтовано необхідність дослідження моделей та методів, створення сучасних навчальних систем на основі штучного інтелекту для впровадження їх при автоматизації створення інтелектуальних систем безперервного навчання, а саме методів перетворення в алгебрі предикатних операцій, розробки бази логічних навігаційних правил адаптації контенту та адаптивного приховування посилань [3].

Відповідно до визначеного комплексу методів і технологій інтелектуальних гіпермедіа систем встановлено можливості їх застосування для потреб адаптивного навчання. У роботі показано переваги використання комбінації з декількох методів та технологій для створення інтелектуальної адаптивної навчальної гіпермедіа системи, яка охопить усі аспекти навчання та допоможе користувачеві в повній мірі оволодіти навчальним матеріалом, гарантуючи повні знання з певної дисципліни [4].

ВИРІШЕННЯ ЗАДАЧІ

Основою для дослідження математичного апарату з моделювання процесів обробки даних, як основи побудови інформаційної технології обрано алгебру скінченних предикатів, проаналізовано основні моделі представлення знань, розглянуто основні методи ідентифікації знань (класифікація, метод компараторного аналізу). При побудові комплексної моделі інтелектуальної адаптивної навчальної гіпермедійної комп'ютерної

системи визначено математичний інструментарій на базі алгебри скінченних предикатів для представлення знань та моделювання стратегії навчання в інтелектуальних гіпермедіа системах з елементами адаптації до моделі M_1 користувача [5].

Алгебра предикатів описує тільки знання про факти. Алгебра предикатних операцій формалізує операції над знаннями, що представлені у вигляді відношень на деякому предметному просторі M . Алгебра предикатів описує декларативну складову знань, а алгебра предикатних операцій – процедурну складову адаптації навчання. Введено алгебру предикатних операцій з константами і змінними, тобто різновид алгебри предикатних операцій з базисними елементами, що складаються із «тотожних предикатних операцій» та «константних предикатних операцій». Тотожною предикатною операцією за змінною $X_j (j = 1, \dots, n)$ називається операція $F(X_1, X_2, \dots, X_j, \dots, X_n) = X_j$ при будь-яких $X_1, X_2, \dots, X_j, \dots, X_n \in M$. Ця операція називається операцією вибору аргументу. Існує n таких операцій. Константною предикатною операцією називається операція $F(X_1, X_2, \dots, X_j, \dots, X_n) = P$ при будь-яких $X_1, X_2, \dots, X_j, \dots, X_n \in M$, де P – предикат на множині M . Кожному предикату $P \in M$ відповідає своя константна предикатна операція.

Алгебра предикатів і предикатних операцій у теорії інтелекту може використовуватися для опису баз даних і баз знань та моделювання навчання та опису правил побудови його стратегії. Таким чином, використовуючи алгебру предикатів і предикатних операцій, можна створити інтегровану модель навчання, засновану на традиційних моделях подання знань, а також на моделях подання знань природною мовою [6].

РОЗРОБКА АЛГОРИТМІВ ПОШУКУ НАЙКОРОТШИХ ДИЗ'ЮНКТИВНИХ ФОРМ

Побудова гіпермедійного простору інтелектуальної системи потребує опису зв'язків та вузлів за допомогою апарату алгебри скінченних предикатів, при цьому основним критерієм оптимальності є кількість зв'язків між вузлами. Це відповідає завданню мінімізації площі відповідної характеристичної матриці, тобто добутку числа рядків на число стовпців даної матриці. Число



певних елементів усередині матриці (елементів впізнання) прийом як додаткового критерію оптимізації. Таким чином, будемо будувати найкоротші ДНФ скінченних предикатів і одночасно враховувати вимоги мінімальності числа впізнань [7].

Іншими словами, будемо шукати мінімальне покриття одиничної області N_f скінченного предиката f максимальними інтервалами. Тому одинична область N_f подана у вигляді характеристичної матриці. У цьому випадку завдання мінімізації ДНФ скінченного предиката можна інтерпретувати як задачу максимального стиснення матриці, при якому деякі сукупності рядків цієї матриці, що утворюють інтервали гіперкуба M^n , замінюються відповідними слабо визначеними векторами. Якщо деяка точка u одиничної області N_f належить лише одному з максимальних інтервалів цієї множини, а саме – деякому інтервалу u , то будь-яке найкоротше покриття області N_f буде містити інтервал u . Будемо називати елемент A визначальним, а інтервал u , що однозначно ним визначається – обов'язковим [8].

Щоб встановити чи є деяка точка A визначальною, досить знайти всі сусідні з нею точки з області N_f , а потім побудувати мінімальний інтервал гіперкуба M^n , що їх містить, званий мінімальним поглинальним інтервалом. Якщо всі точки цього інтервалу належать до одиничної області, то цей інтервал є інтервалом області N_f , максимальним і обов'язковим, а елемент A є визначальним. Мінімальний поглинальний інтервал для заданої множини точок простору M^n отримують у такий спосіб. Будується вектор, компоненти якого приймають значення однойменних компонент координат точок з M^n , якщо ці значення збігаються, і приймають значення «-», якщо серед значень однойменних компонент координат точок зустрічаються різні літери з алфавіту. Якщо інтервал u містить багато елементів, то перевірка питання, чи міститься інтервал u в області N_f , зводиться до наступного.

З характеристичної матриці виділяється вектор, утворений перерахуванням рядків, сусідніх вектору u за допомогою стовпців, в яких вектор u має значення «-», а потім виділяється перевірка мінору на виродженість (мінор назвемо виродженим, якщо він містить рядки, елементи яких представляють усі можливі комбінації з

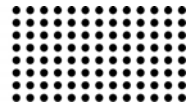
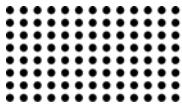
букв вихідного алфавіту). Якщо мінор вироджений, то інтервал u міститься в N_f .

Обчислення обов'язкових інтервалів прискорюється при малій кількості сусідів розглянутої точки. Якщо точка не має сусідів, то вона сама утворює обов'язковий інтервал. Якщо точка має двох сусідів, один з яких відрізняється від A за компонентою i , а інший за компонентою j , то обчислення зводяться до перевірки наявності в області N_f елемента, відмінного від A за двома компонентами i та j . Знайшовши в області N_f принаймні один обов'язковий інтервал, ми починаємо послідовно будувати елементи покриття одиничної області, яке за сприятливих умов дасть точне розв'язання нашої задачі.

Розглянемо поточну ситуацію, коли частина елементів одиничної області N_f вже покрита деякими інтервалами, а інша частина, що утворює поточну, тобто змінювану в процесі розв'язання множину N^* , складається з елементів одиничної області, які ще не належать жодному з вибраних інтервалів. Нехай деякий елемент C поточної множини N^* належить одразу декільком максимальним інтервалам області N_f . Розглянемо перетини цих інтервалів з множиною N^* , називаючи їх надалі проєкціями інтервалів на множину N^* . Якщо серед цих проєкцій знайдеться така, яка містить в ролі своїх підмножин інші, то можна вважати обґрунтованим вибір будь-якого відповідного їй максимального інтервалу (їх може виявитися декілька) як чергового елемента найкоротшого покриття. Цей інтервал забезпечує покриття всіх елементів з N^* , які тільки можна покрити інтервалами множини N_f , які покривають одночасно елемент C . Такий інтервал назвемо допустимим, відзначимо при цьому, що даний інтервал виявляється допустимим лише при даній послідовності попередніх виборів інтервалу покриття.

Тепер розглянемо спосіб знаходження визначальних елементів в проміжних ситуаціях. Нехай деякі два елементи одиничної області N_f несумісні, якщо поглинальний мінімальний інтервал міститься в N_f . Елемент C виявляється визначальним, якщо поглинає інтервал для всіх елементів множини N^* , сумісних з елементом C , міститься в області N_f . [9]

Вибір у множині N^* , виконуваний під час пошуку сумісних з C елементів, прискорюється тим, що він обмежує розгляд елементів, що належать до мінімального поглинального інтервалу для всіх сусідів елемента C , як



сумісні з C елементи можуть перебувати тільки в цьому інтервалі. Обидва мінімальних поглинальних інтервали як для сусідів елемента C , так і для сумісних з ним елементів будуються легко. Нарешті, вищенаведеним способом встановлюється, чи міститься останній з цих інтервалів одиничної області N_i . Позитивна відповідь на це питання доводить, що елемент C є визначальним. Допустимим інтервалом в цьому випадку виявляється будь-який мінімальний інтервал області N_i , що містить знайдений мінімальний поглинальний інтервал для сумісних з C елементів. Зокрема, ці два інтервали можуть співпасти.

Проміжні інтервали суттєво відрізняються від початкового тим, що в початковій ситуації від визначального елемента C потрібно, щоб в одиничній області N_i містився мінімальний поглинальний інтервал для всіх сумісних з C елементів області N^* , в той час як у поточній ситуації ця вимога послаблена: там розглядаються мінімальні поглинальні інтервали тільки для тих елементів, сумісних з визначальним елементом, які належать до поточної множини N^* .

Ця відмінність визначає ланцюговий характер процесу побудови найкоротшого інтервального покриття одиничної області N_i . Включення в покриття чергового допустимого інтервалу супроводжується відповідним скороченням множини N^* , а це в свою чергу призводить до того, що деякі з решти елементів в N^* можуть стати визначальними, що дозволить включити в розв'язання нові допустимі інтервали, далі скоротити поточну множину N^* і т.д. У розглянутому способі конструювання найкоротшого інтервального покриття одиничної області N_i можуть зустрітися критичні ситуації, коли в поточній множині N^* не вдасться знайти жодного визначального елемента.

У такому випадку для отримання точного розв'язання потрібно виконати перебір різних варіантів побудови покриття, що відповідають вибору різних максимальних інтервалів як чергового елемента покриття. Необхідно вжити заходів щодо скорочення цього перебору, застосувавши розгалужені процеси. Замість допустимого інтервалу, в описаній критичній ситуації вибираємо деяку сукупність інтервалів, тобто множину максимальних інтервалів, щодо якої відомо, що в ній міститься, принаймні, один з допустимих в поточній ситуації інтервалів, хоча і невідомо, який саме. При

цьому поняття допустимого інтервалу вживається в широкому сенсі. Назвемо допустимою таку сукупність інтервалів одиничної області N_i , якщо існує деяке найкоротше інтервальне покриття області N_i , що її містить. Якщо сформована на поточний момент часу частина покриття, що конструюється є допустимою, то деякий максимальний інтервал буде допустимим в даному випадку, якщо, прийнявши його як черговий елемент покриття, ми отримаємо знову допустиму сукупність. Розгляд різних варіантів вибору чергового елемента покриття з достатньої сукупності утворює точку розгалуження процесу пошуку найкоротшого інтервального покриття. Число вихідних з цієї точки гілок дорівнює числу елементів у даній достатній сукупності [10].

Скорочення перебору зводиться в зв'язку з цим до знаходження в поточній ситуації деякої мінімальної достатньої сукупності. Наприклад, достатньою є сукупність всіх максимальних інтервалів, яким належить певний елемент C області N_i . Достатньою в поточній ситуації виявляється сукупність максимальних інтервалів, до яких належить певний елемент C множини N^* і деякі мають різні, що не містяться оди в одній проекціями на множині N^* . Число елементів в цих допустимих сумах, як правило, менше, якщо у елемента C , що їх задає мало сусідів.

Можна зробити висновок, що для скорочення перебору під час пошуку оптимального розв'язання даної задачі: попередньо зважити всі елементи N_i , визначаючи його вагу як число сусідів, а наступні перебори в області N_i організувати в порядку не спадання цих ваг.

Даний метод породжує наближені методи, в яких не обов'язково розглядати всі варіанти, пов'язані з вибором різних елементів достатніх сукупностей. Наприклад, в кожній точці розгалуження може вибиратися один шлях, «кращий» за певним критерієм. Також може бути спрощено сам спосіб утворення точок розгалуження, в результаті чого задача знаходження мінімальної достатньої сукупності може вирішуватися також наблизено або навіть не ставиться.

В описаному способі аналізу характеристичної матриці області N_i скінченного предиката f , що мінімізується необхідно розглядати велику кількість різних сполучень елементів у множині N_i . Наприклад, відшукуючи в області N_i сусідів, ми повинні перевірити всі пари елементів цієї множини. Цей перебір можна значно скоро-



тити, задаючи розглянутий скінченний предикат f безпосередньо на гіперкубі M^n , елементи якого впорядковані і певним чином пронумеровані. Якщо в деякій вершині гіперкуба M^n скінченний предикат f приймає значення 1 , тобто якщо ця вершина належить до одиничної області N_f , то, відшукуючи його сусідів в області N_f , не обов'язково перебирати всі елементи цієї області і порівнювати їх з даними, а досить переглянути лише сусідні елементи гіперкуба M^n (їх число дорівнює n) і вибрати з них ті, на яких предикат f приймає значення 1 . Таким чином, суцільний перебір елементів в характеристичній матриці, відповідної області N_f , замінюється впорядкованим вибором і аналізом елементів гіперкуба M^n . Такий спосіб назовемо скануванням.

Розглянемо метод мінімізації ДНФ скінченного предиката, що базується на виборі в одиничній області елементів в порядку зростання числа сусідів. Скінченний предикат, що мінімізується видається безпосередньо в табличній формі, де в лівій колонці перераховано всі елементи гіперкуба M^n – набори значень аргументів, впорядковано відповідно до позиційного k -го коду, де k – число букв алфавіту A , а праворуч значення на наборах скінченного предиката.

Оскільки послідовність наборів чітко впорядкована, тобто кожен набір однозначно визначено місцем, зайнятим у таблиці, то їх можна не показувати, що і робиться при поданні функції f , що мінімізується в пам'яті комп'ютера, коли набір ототожнюється з адресою, куди поміщається відповідне значення скінченного предиката f . Значення скінченного предиката на сусідніх елементах гіперкуба отримуються зміною значення тієї компоненти адреси, за якою відшукуються сусідні елементи. Чим менше сусідів в області N_f має певний елемент цієї множини, тим меншому, як правило, числу різних максимальних інтервалів він належить. З огляду на це, описуваний метод починає побудову інтервального покриття множини N_f з пошуку інтервалів для покриття елементів з мінімальним числом сусідів – в цьому випадку конструювання покриття буде цілеспрямованим.

Першим етапом є побудова таблиці сусідів, в якій для кожного елемента характеристичної матриці показано, за якими змінними цей елемент має сусідів в одиничній області N_f . Цю таблицю сумісний з таблицею значень скінченного предиката f , заповнюючи прочер-

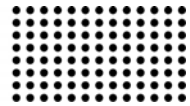
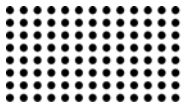
ком відповідні позиції для елементів нульової області. Розгляд починаємо з елементів, які не мають сусідів; якщо таких не виявлено, то вибираємо елементи з одним сусідом. Знаходимо для цих елементів обов'язкові інтервали і далі проводимо всі операції, описані вище в алгоритмі.

ПОБУДОВА АЛГОРИТМІВ МІНІМІЗАЦІЇ ДОВІЛЬНИХ ДИЗ'ЮНКТИВНИХ ФОРМ

У більшості задач скінченний предикат f задається одразу в довільній диз'юнктивній формі, яку можна уявити певною характеристичною матрицею U . Мінімізувати її в цьому випадку можна, скориставшись одним з описаних вище методів, але для цього буде потрібно перейти спочатку до досконалої ДНФ, задавши її в табличному вигляді. Це не завжди практично просто, а при великій кількості аргументів скінченного предиката неможливо. У зв'язку з цим практичний інтерес представляють методи безпосереднього перетворення довільної характеристичної матриці U до деякої мінімальної форми без попереднього отримання СДНФ.

Найпростішими методами мінімізації ДНФ є методи усунення надлишковості в ній, і працюють вони відповідно до визначення поняття безнадлишковості. При цьому передбачається, що безнадлишкова ДНФ буде служити прийнятним наближенням до оптимальної форми. Оптимальною ми будемо як і раніше вважати найкоротшу ДНФ. За визначенням ДНФ безнадлишкова, якщо з неї не можна видалити жодного впізнання в будь-якій елементарній кон'юнкції. Іншими словами, характеристична матриця U безнадлишкова, якщо з неї не можна видалити жодного рядка і немає такого елемента в матриці, відмінного від «-», щоб його значення можна було замінити на «-», отримавши при цьому матрицю, еквівалентну вихідній.

На основі даного визначення введемо операції видалення рядка і видалення літери. Рядок U_i матриці U видаляється, якщо представлений матрицею скінченний предикат при цьому не зміниться. Видаленням літери назовемо операцію зміни значення елемента U_{ij} на «-», якщо це можливо. Рядок U_i можна видалити лише в тому випадку, якщо будь-який інтервал, що поглинається нею в той же час поглинається ще яким-небудь рядком матриці U .



А цю умову можна трансформувати у таке: мінор, утворений перетином рядків, несусідніх рядку U_i має значення «-», повинен бути вироджений.

Дійсно, якщо цей мінор не вироджений, то знайдеться деякий сусідній йому інтервал. Замінивши значеннями його компоненти значення «-» відповідних компонент рядка U_i , отримаємо інтервал, що поглинається рядком U_i і не поглинається більше жодним рядком характеристичної матриці U . В цьому випадку рядок U_i видалити не можна. Якщо ж мінор вироджений, то не існує такого інтервалу, який поглинався б тільки рядком U_i .

Замінити буквене значення деякого елементу U_{ij} матриці U на «-» можна тільки в тому випадку, якщо буде вироджений мінор, утворений перетином тих рядків, які не є сусідніми інтервалу, сусідньому рядку U_i за компонентою j , з тими стовпцями, в яких рядок U_i має значення «-». Справді, в цьому випадку зазначений сусідній вектор можна додати в матрицю U_i , склеївши його з рядком U_i , отримати необхідний результат.

Користуючись описаними операціями, ми завжди можемо знайти безнадлишкову ДНФ, виходячи з деякої довільної. При цьому слід пам'ятати, що отримуваний результат залежить від порядку вибору як операцій, так і рядків характеристичної матриці U , до яких вони застосовуються. Багатокроковий процес спрощення може мати ланцюговий характер. У тих випадках, коли описаний спосіб спрощення ДНФ виявляється трудомістким, можна спростити ДНФ спочатку більш швидкодієвим способом, а потім вже усувати надмірність отриманого результату.

Наприклад, досить ефективною виявляється така процедура: у характеристичній матриці U відшукуються пари суміжних рядків і до них застосовується операція узагальненого склеювання. Якщо результуючий інтервал поглинає деякий з цих рядків (або обидва), то він її замінює, в іншому випадку не використовується. У багатьох практичних ситуаціях завдання побудови найкоротшої ДНФ, може бути скорочена шляхом знаходження в останній обов'язкових членів, які повинні бути наявні в будь-якій найкоротшій ДНФ з безнадлишковими членами і навіть в будь-якій безнадлишковій ДНФ. У характеристичній матриці, яка задає скорочену ДНФ, в цьому випадку відшукуються відповідні обов'язкові рядки.

Рядок U_i матриці U є обов'язковим, тобто має бути неодмінно включений у розв'язання, якщо його не можна викинути з самого початку. Тобто він є обов'язковим, якщо мінор, утворений перетином інших рядків, не сусіднього U_i , за допомогою стовпців, де він має значення «-», що не вироджений. Сукупність обов'язкових рядків складає ядро розв'язання.

З іншого боку, рядок матриці, що покривається ядром, не зможе належати жодному з мінімальних покриттів, жодному з безнадлишкових. Такий рядок задовольняє умові: мінор, утворений перетином рядків, що належать ядру і не сусідніх цьому рядку, за допомогою стовпців, в яких він має значення «-», вироджений. Сукупність таких рядків назовемо антиядром розв'язання. Ці рядки можна відразу виключити з розгляду, видаливши з матриці U .

В результаті знаходження ядра і антиядра розв'язання виконуваний під час пошуку оптимального покриття перебір може значно скоротитися. Припустимо тепер, що вихідна характеристична матриця U задає не скорочену, а деяку безнадлишкову ДНФ скінченного предиката f . Запитання, чи можна в цьому випадку виявити ядро розв'язання безпосередньо в матриці U , не вдаючись до попереднього отримання множини всіх простих імплікант шляхом відповідного розширення матриці U ?

Перш за все, зауважимо, що оскільки обов'язкові рядки повинні міститися в будь-якій безнадлишковій ДНФ, то всі вони, очевидно, містяться і в даній характеристичній матриці. Залишається розпізнати їх там.

Раніше було показано, що максимальний інтервал U_i одиничної області N_f обов'язковий, якщо він володіє деяким елементом, що не має в множині N_f таких сусідів, що не належали б до цього інтервалу. Очевидно, що такі сусідні інтервали могли б перебувати тільки в таких максимальних інтервалах, які суміжні з інтервалом U_i . Рядок U_i безнадлишкової матриці U обов'язковий в тому і тільки в тому випадку, коли не буде виродженим мінор, утворений перетином рядків, суміжних з рядком U_i , зі стовпцями, в яких рядок U_i має значення «-».

Дійсно, якщо цей мінор вироджений, то для будь-якого елемента a інтервалу U_i в матриці U знайдеться такий інтервал U_j , який містить деякого його сусіда (якщо U_j і U_i суміжні). Це означає, що елемент a нале-



жити, принаймні, двом максимальним інтервалам – один з них може бути отриманий узагальненням інтервалів U_i і U_j .

Якщо ж мінор не вироджений, то знайдеться сусідній йому інтервал. Присвоївши значення його компонент відповідним компонентам рядка U_i , ми отримаємо елемент інтервалу U_i , не має в одиничній області U_i сусідній за межами цього інтервалу, що і доводить обов'язковість рядка U_i .

Повернемося тепер до ситуації, в якій характеристична матриця U представляє множину S всіх максимальних інтервалів одиничної області N_i , і розглянемо задачу вибору з неї такої допустимої сукупності максимальних інтервалів, для якої відповідно до визначення знайдеться найкоротше інтервальне покриття області N_i , що її містить.

Представимо цей вибір як багатокроковий процес, при якому поточна ситуація характеризується розбиттям множини S на три частини: деяку вже побудовану сукупність S^+ , множину S^- , щодо якої відомо, що існує деякий найкоротше інтервальне покриття області N_i , що містить S^+ і не перекресне з S^- , і поточний залишок S^* , який визначається як $S \setminus (S^+ \cup S^-)$. Усі три множини є змінними, причому в ролі початкового значення множини S^+ може бути прийняте ядро розв'язання, що складається з обов'язкових інтервалів, а в ролі початкового значення множини S^- – антиядра.

Черговий крок полягає у додаванні у множину S^- деякого елемента, обраного з S^* , причому такого, щоб нове значення множини S^- також мало описану вище рису. Ця операція супроводжується, якщо це можливо, розширенням множини S^+ , а також відповідним скороченням множини S^* .

Процес триває до деякої тупикової ситуації, коли в множині S^* вже не вдається знайти елементів з необхідними властивостями. Тоді виникає необхідність в організації перегляду різних варіантів продовження побудови інтервального покриття області, або у виборі будь-якого конкретного варіанту, перспективного, але такого, що не гарантує оптимальність отриманого при його реалізації результату.

Позначаючи як і раніше, через N_i множину елементів з N_i , не покритих обраною допустимою сукупністю S^+ , будемо говорити, що інтервал U_j поглинає інтервал U_i на множині N^* , якщо будь-який елемент з N^* , що

належить інтервалу U_i , належить інтервалу U_j . В основу методу побудови допустимої сукупності покладене таке твердження: інтервал U_i може бути переданий з множини S^* у множину S^- , якщо в множині S^* існує деякий інший інтервал U_j , який поглинає U_i на множині N^* . Умови можливої передачі інтервалу U_i з множини S^* у множину S^+ залишаються незмінними: він не повинен покриватися сукупністю інших інтервалів, що належать об'єднанню множин S^* і S^+ .

Оскільки при передачі деякого інтервалу з множини S^* у множину S^- відбувається скорочення множини S^* , це може дозволити знайти потім в множині S^* інший інтервал, який можна передати множині S^+ . У свою чергу передача деякого інтервалу у множину S^+ може створити умови для подальшого скорочення поточного залишку S^* . В цьому і полягає ланцюговий характер описуваного процесу.

Представляючи множини S^+ , S^- і S^* в матриці U , будемо розбивати її в поточній ситуації на три частини – рядкові мінори, що позначаються відповідно через U^+ , U^- і U^* . В процесі побудови допустимої сукупності матриці U^+ і U^- будуть рости, а матриця U^* – скорочуватися.

Рядок U_i матриці U^* можна передати в матрицю U^+ , якщо в матриці U^* залишиться деякий інший рядок U_j такий, що буде виражений мінор, утворений перетином несусідніх рядку U_i рядків матриці U^+ і U^* зі стовпцями, в яких рядок U має значення «-». Продовжуючи мінімізацію розглянутого скінченного предиката $f(x_1, x_2, \dots, x_n)$, можна отримати в явному вигляді одиничний інтервал N_i , видалити з нього елементи, що покриваються знайденою допустимою сукупністю S^+ , побудувати матрицю бінарного відношення приналежності елементів, що залишилися інтервалах з множини S^* і приступити до пошуку її найкоротшого покриття. Однак при великій кількості елементів в області N_i цей метод виявляється трудомістким, а матриця, що будується – громіздкою.

МЕТОД, ЗАСНОВАНИЙ НА ВИКОРИСТАННІ МАТРИЦІ ПРОСТИХ СУКУПНОСТЕЙ

Так називають матрицю, що отримується з імплікантної таблиці видаленням рядків, що поглинають деякі з рядків, що залишаються в матриці. Імплікантна таблиця, як було визначено вище, задає бінарне відношення приналежності елементів одиничної області N_i максимальним інтервалам цієї області, причому рядки

відповідають елементам, а стовпці – інтервалам. Кожен рядок матриці простих сукупностей задає одиницями у відповідних стовпцях деяку сукупність, тобто таку сукупність S_i максимальних інтервалів одиничної області N_i , яка має такі властивості, що впливають з описаного способу отримання матриці:

а) в одиничній області існує певний елемент, що належить кожному з інтервалів сукупності S_i і не належить більше жодному з максимальних інтервалів;

б) не існує іншої сукупності S_j , що міститься в S_i ($S_j \subset S_i$) і має властивість а).

Будь-яке безнадлишкове інтервальне покриття містить, принаймні, по одному інтервалу з кожної простої сукупності. Матриця простих сукупностей може бути, тому інтерпретована як перелік відповідних вимог, що пред'являються до шуканого розв'язання поряд з вимогою мінімальності, причому число цих вимог мінімізоване шляхом виключення завідомо зайвих, автоматично виконуваних при задоволенні решти вимог.

Сформулюємо таке правило, що дозволяє знаходити прості сукупності безпосередньо в множині S простих імплікант, що задаються відповідними максимальними інтервалами одиничної області. Підмножина S_i множини S являє собою сукупність в тому і тільки в тому випадку, коли:

а) перетин всіх інтервалів з S_i не міститься в об'єднанні інтервалів з $S \setminus S_i$;

б) в множині S не існує іншої підмножини S_j , що міститься в S_i і, що має властивість а);

Це правило легко переноситься на випадок, коли множина S задана характеристичною матрицею U . Оскільки непорожній перетин інтервалів також є інтервалом. Пошук простих сукупностей проводиться в порядку зростання потужності розглянутих підмножин. Причому, при виявленні певної простої сукупності все підмножини, що її містять виключаються з подальшого розгляду. Якщо з множини простих імплікант попередньо виділені допустима сукупність S^+ і поточний залишок S^* , то пошук простих сукупностей обмежується множиною S^* . Аналізу піддаються тільки підмножини цієї множини. Однак при побудові відповідних мінорів використовуємо рядки як матриці S^* , так і матриці S^+ .

Також для кожного окремого елемента подання інформації в гіперструктурі – навчальному предикату введемо відношення приналежності Δ відібраних

навчальних документів t_a, t_b одному конструктиву в такий спосіб:

$$t_a \Delta t_b \Leftrightarrow (\forall p \in P)(R(t_a, p) = R(t_b, p)) \quad (5)$$

Відношення Δ визначає мінімально розподілене подання цілісності. Відношення Δ має властивості рефлексивності $t_a \Delta t_b$; симетричності $(\forall t_a, t_b \in T)(t_a \Delta t_b) \Rightarrow t_a \Delta t_b$; транзитивності $(\forall t_a, t_b, t_c \in T)(t_a \Delta t_b)(t_b \Delta t_c) \Rightarrow t_a \Delta t_c$, отже, відношення Δ є еквівалентністю.

Для понять предметної галузі, що лежать в основі побудови гіперструктури, застосовано відношення Π приналежності понять p_i, p_j понятійній основі конструктиву P :

$$p_i \Pi p_j \Leftrightarrow (\forall p \in T)(R(t, p_i) = R(t, p_j)) \quad (6)$$

Відношення Π визначає закономірності структуризації цілісності. При розробці елементів алгебри предикатів та предикатних операцій побудований ряд предикатів і доведена їхня функціональність. Предикат E_Δ пропонується використовувати для визначення дидактичної близькості документів t_a і t_b з множини однотипних форм: якщо $E_\Delta(t_a, t_b) = 1$, тоді $R(t_a, p) = R(t_b, p)$ для будь-якого поняття p з множини понять P . Отже, всі властивості документів t_a і t_b , що відповідають поняттями з множини Ψ , збігаються. Якщо $E_\Delta(t_a, t_b) = 0$, то таке поняття $p \in P$, для якого $R(t_a, p) \neq R(t_b, p)$, доводить дидактичне розходження t_a і t_b . Аналогічно предикат $E_\Pi(p_i, p_j) = 0$ може бути використаний для визначення функціональної еквівалентності понять p_i і p_j з множини P : якщо $E_\Pi(p_i, p_j) = 1$, тоді $R(t, p_i) = R(t, p_j)$ для будь-якого документа t з множини T , тобто ці поняття одночасно відповідають



документу $t \in T$, або одночасно не відповідають. Обидва введені предикати E_{Δ} і E_{Π} є еквівалентно-стями, отже, факторизують множини різних форм і понять. Предикат E_{Δ} визначає розбивання множини T на шари S дидактично близьких документів. Предикат E_{Π} визначає розбивання множини P на шари L функціонально еквівалентних понять; поняття з різних шарів L функціонально еквівалентними не є.

При формуванні адаптивної моделі M_2 , генерація стратегії навчання S відбувається за допомогою навігаційних правил шляхом порівняння поточної моделі M_{1i} користувача з еталонною моделлю курсу M_{ei} на i кроці за допомогою введеного у роботі коефіцієнта толерантності знань. Ступінь толерантності знань $S(M_1, M_e)$ – це відношення потужності множини M_1 елементів знань відтвореного образу еталона знань M_{1i} на i -ому кроці навчання до потужності множини M_e елементів знань еталона знань M_{ei} .

$$S(M_1, M_e) = \frac{|M_{1i}|}{|M_{ei}|} \quad (10)$$

Значення ступеня толерантності $S(M_1, M_e)$ перебуває в інтервалі $[0, 1]$. Ті знання, які відтворюються, порівнюються з еталоном. У процесі порівняння двох моделей з множини навчальних дій I формується підмножина дій \tilde{I} ($\tilde{I} \in I$), вивчення яких необхідне для успішного навчання. На цьому процес формування M_2 закінчується і починається процес навчання відповідно до S_i , який триває до так званої «контрольної смуги» (тип I_j), після чого здійснюється перехід на наступний рівень ітерації з модернізацією моделі M_1 і адаптацією під неї моделі M_2 . Процес триває до досягнення необхідного рівня засвоєння знань суб'єктом навчання. При цьому користувач одержує можливість самостійно вивчати матеріал під управлінням технології адаптивної гіпермедіа,

Ця модель заснована на використанні методу компараторної ідентифікації для розбивання навчального матеріалу на класи еквівалентності і зв'язування для побудови гіперструктури дидактичних матеріалів, яка в свою чергу, надає і контролює проходження навчальних матеріалів в залежності від коефіцієнта толерантності знань та навігаційних правил, що представлені рівняннями алгебри скінчених предикатів.

ВИСНОВКИ

Найбільш поширеним підходом до адаптивної навігації є різні прийоми адаптації посилань шляхом визначення їх «корисності». Розроблено також методи та алгоритми, що дозволяють упорядкувати вузли гіперпростору з метою отримання зв'язного тексту (мікрорівень), отримання послідовності навчальних впливів із ускладненням викладу досліджуваних понять (макрорівень), вибору послідовності контролюючих впливів.

Для представлення методичних знань доречно застосування правил в базисі алгебро-логічних рівнянь, що визначають вид навчальних впливів, їх рівень складності та порядок надання в залежності від попередніх успіхів суб'єкта навчання. Відповідно до обґрунтування основних вимог до методу навчання та відновлення знань в галузі дослідження методів штучного інтелекту запропоновано метод навчання та відновлення знань, який передбачає етапи сприйняття еталону знань, формування, відтворення суб'єктивного образу та порівняння відтвореного образу до еталону знань, що дозволило підвищити ефективність інтелектуальних інформаційних систем навчання.

Використовуючи алгебру скінчених предикатів як математичний апарат, обґрунтовано та побудовано багаторівневу модель організації гіпермедійного простору, введено поняття навчальних предикатів, розроблено методи навчання та відновлення знань, які дозволяють підвищити інтенсивність та ефективність комп'ютеризованого навчання та процесу створення комп'ютерних навчальних гіпермедіа просторів і віртуальних середовищ з елементами штучного інтелекту, що сприятиме спрощенню сприйняття користувачем навчального матеріалу.



REFERENCES

1. Bondarenko M. Teoriya Intellekta / M.F. Bondarenko, Yu.P. Shabanov-Kushnarenko. Kharkiv.: «CMIT», 2007. – 221 p.
2. Bondarenko M. Algebra Predikativ i Predikatnykh Operatsiy / M. Bondarenko, Z. Dudar, Yu.P. Shabanov-Kushnarenko // Radioelectronics & Informatics. – Kharkiv: KhNURE, 2005. – № 1. – pp. 80–87
3. Tolkiehn G., Makariti A. MyWebRTC, a Free Do-It-Yourself Kit for Secure Real-Time Internet-Communication //Information Technologies in Information Business Conference (ITIB 2015) – Kharkiv, 2015. – pp 81-84.
4. Kuzikov B. Programne Zabezpechennya Dlya Pobudovy Individualnih Navchalnyh Traektorij / B.O.Kuzikov // Informatika, Matematika, Avtomatika : Materialy ta programma naukovy-tekhnichnoi konferencii, 2012, Sumy. –: SumDU , 2012. –pp. 54-55.
5. Shubin I., Karmanenko O., Umyarov K. The Methods of Adaptation in Computer-Based Training Systems //Information Technologies in Information Business Conference (ITIB 2015) – Kharkiv, 2015. – S 64-67
6. Sharonova N. Programmaya Sistema Avtomaticheskogo Referirovaniya Teksta / N.V. Sharonova, A.B. Kovanev, Z.A. Aliseyko // Vestn. Kherson. Nats. Tekhn. Un-ta. – 2005. – № 1(21). – pp. 331–334.
7. Kuzikov B. Adaptivnaya Sistema Distantsionnogo Obucheniya Dlya Razvitiya Dostupa k Kachestvennomu Obrazovaniiyu: Opyt SumGU / V.A.Lyubchak, B.O.Kuzikov, T.V.Lavrik// Nove Informatsionnye Tekhnologii v Obrazovanii Dlya Vseh/ kollektiv avtorov. – K. : Akademperiodika, 2012. –pp.167-182.
8. Kabassi K. Using Web Services Personalised Web-based learning / K. Kabassi, M. Virvou // Educational Technology & Society. – 2003. –6(3). – pp. 61–71.
9. Shubin I., Kirichenko I. Informatsionnye Tekhnologii Modelirovaniya Adaptivnykh System Obucheniya // Matherials of The 5Th Mezhdunarodnoy nauchno-tehnicheskoy konferencii «Informatsionnye sistemy i tekhnologii ICT-2016», Kharkiv, 2016 – S. 244-245.
10. Shubin I., Kirichenko I. Modeli Intelktualnoy Podderzhki Navigatsii v Computer Ttraining Systems / International Journal “Information Models and Analyses” Vol. 2/ 2013, Number 2. – Bulgaria – 2013 – pp. 194-199

Рецензент: д.т.н., проф. Соколова Н.А.,
Херсонський національний технічний університет