УДК 004.4’232

Я. Драган[1], В. Овсяк[2], О. Овсяк[3]
[1]Національний університет “Львівська політехніка”,
кафедра програмного забезпечення
[2]Українська академія друкарства,
[3]Київський національний університет культури і мистецтв, Львівська філія

МЕТОДОЛОГІЯ СИНТЕЗУ МОДЕЛЕЙ АЛГОРИТМІЧНОЇ СКЛАДОВОЇ АВТОМАТІВ

Описано етапи синтезу, оптимізації і дослідження математичних моделей алгоритмічної складової автоматів методом алгебри алгоритмів, декомпозиції і математичної індукції.

Ключові слова: метод, методологія, модель, інформаційна технологія, декомпозиція, секвентуння, елімінування, паралелення, функційний унітерм, унітерм.

There is describe by the stages of synthesis, optimization and study mathematical models of machine algorithmic component by the methods of algorithm algebra, decomposition and mathematical induction..

Key words: method, methodology, model, information technology, decomposition, sequentization, elimination, parallelization, function uniterm, uniterm.

Вступ і формулювання задачі

Розвиток технологій і практична необхідність виконання великих розрахункових робіт під час конструювання атомної бомби, приховування і перехоплення відомостей (криптографія), автоматизації керування вогнем протиавіаційного захисту сприяли започаткуванню кібернетично-інформаційної доби [1]. В основі її є концепція автомата, сформульована у працях Мак-Калока – Пітса [2, 3]. Формально автомат (A) є п’ятіркою величин (V, S, W, φ, ψ), де V – дії (впливи), S – стани, W – реакції автомата, φ – функція переходів, яка є відображенням (→) декартового добутку ($V\times S$), що записується як $V\times S\rightarrow S$, ψ – функція виходів як відображення $S\times V\rightarrow W$. Алгебра автомата є триосновною; основи її утворюють множини V, S і W.

В.Глушков створив абстрактну модель електронної обчислювальної машини у вигляді двох автоматів, а власне керуючого і операційного з двосторонніми зв’язками між ними [4]. Для побудови математичної моделі автомата В.Глушков розробив систему алгоритмічних алгебр [4]. Утворена вона двома алгебрами, а саме алгеброю операторів та алгеброю умов. Алгебра операторів складається з операторів композиції (множення (*) – послідовного застосування операторів), α-диз’юнкції (альтернативи), диз’юнкції, фільтра і циклу. Алгебра умов містить узагальнені на тризначний алфавіт операції булевої алгебри і запроваджену нову операцію прогнозування (лівого множення умови на оператор). Г.Цейтлін модифікував систему алгоритмічних алгебр В.Глушкова, зокрема введені операції контрольної точки і синхронізування [5].

Використання двох алгебр (операторів і умов) ускладнює синтез автоматів. Крім того, операція композиції є асоціативною (див. с. 126 [4]). Однак в автоматах послідовне застосування операторів (властивість асоціативності) може мати місце тільки у часткових випадках. Для того, щоб переконатися у цьому, розглянемо тривіальний приклад. Припустимо, що маємо три змінні: x=0, y=0 і z=0, над якими виконуються такі дії: x=5+5, y=x:2 та z=y-1. Якщо спочатку виконати ці дії у такій послідовності: ((x=5+5)*(y=x:2))*(z=y-1), а потім – у такій: (x=5+5)*((y=x:2))*(z=y-1)), то отримаємо значення, які відрізняються.

Відома одноосновна алгебра алгоритмів [6] та її модифікація [7] з неасоціативною операцією секвентування, засобами яких виконується опис автоматів.

Автомати можуть бути реалізовані апаратно або апаратно-програмно. Найчастіше типовою реалізацією є якраз апаратно-програмна. До того ж досягнуті успіхи інтегральних напівпровідникових технологій забезпечують належний рівень функціонування апаратних засобів автоматів. Проблематичнішим є створення алгоритмічного і програмного забезпечення автоматів. Тому нижче розглянуто методологію застосування алгебри алгоритмів для побудови математичних моделей програмних складових автоматів, а власне інструментальних засобів комп'ютерних систем. Ця методологія таким самим способом може бути також використана і для синтезу апаратних складових автоматів.

Моделювання інформаційних технологій і систем

Основою інформаційних технологій є теорія автоматів, а з часів В.Глушкова утвердилась концепція, що мовою її є відповідна версія сучасної алгебри. Для неї ж характерне те, що в центрі уваги стають властивості операцій, а не об'єкти, над якими їх виконують, що своєю чергою призвело до поняття алгебричної структури та алгебричної системи як множини з означеною на ній сукупністю операцій і стосунків, які сформувались у середині XX ст. Одним з основних напрямків цієї теорії є вивчення стосунків між загальними властивостями класів алгебричних систем та синтаксичними властивостями мови, якою означують ці структури [3].

Оскільки апаратом теорії алгоритмів є різного роду варіянти алгебр, то природно скористатись для розроблення відповідних варіантів теорії так званим аксіоматичним методом. Хоч при запровадженні його в основи математики надії Д.Гільберта, що він може стати підставою всієї математики – усіх її розділів, не оправдались, він може бути корисним (з належною модифікацією) сьогодні. Для того пригадаймо, що аксіоматичний метод – це спосіб побудови наукової теорії (головно в математиці), коли в основу побудови покладено певні твердження, звані аксіомами, а всі решти факти отримують як логічні висліди з аксіом (дедукція). І тоді постають не тільки проблеми – звідки взяти аксіоми, але й проблеми несуперечності, повноти і незалежності цих аксіом. А як відомо з концепцій К.Геделя неповноти аксіоматичних теорій аксіоматичний метод не може бути взятий за остаточну основу обґрунтування теорії, проте він може стати засобом обґрунтування від супротивного несуперечності й повноти системи аксіом, коли доводять не саму теорему, а рівносильну їй протилежну до зворотної та коли вислід змінюють на супротивний і показують суперечливість засновкові (reduction ad absurdum – зведення до абсурду), або ж використовують поняття моделі, чи інакше , інтерпретації, коли кожному початковому поняттю й стосункові цієї тестованої теорії ставиться у відповідність певний поля інтерпретації, яка теж має свою теорію, що може бути аксіоматичною й дає підстави установлення відносних несуперечності, повноти і незалежності [3, 8].

Якщо можна побудувати модель, як її трактують у математичній логіці, тобто об'єкт, на якому реалізуються аксіоми системи, то цю систему post factum трактують, що вона має названі властивості. Історично подібну функцію виконала в обґрунтуванні геометрії Лобачевського-Бойяї інтерпретація А.Пуанкаре, реалізуючи аксіоми геометрії площини Лобачевського (гіперболічної геометрії) на одиничному крузі $E=\{z:|z|\leq 1\}$ площини комплексної змінної. Ф. Кляйн та А.Пуанкаре установили несуперечність неевклідової геометрії щодо евклідової, а несуперечність її Д. Гільберт звів до несуперечності арифметики [3, 8].

Наведені факти підтверджують тезу, що є два принципово різні тлумачення терміну математична модель: 1) це об'єкт відповідного розділу математики, у структурі якого в стислій конструктивній формі втілені істотні для розв'язування цього класу задач властивості досліджуваних об'єктів [9] – так прийнято у математиці, а відтак, у всіх технічних науках і 2) це вже описана інтерпретація реалізацій аксіом, як прийнято в логіці та основах математики. Наведене підтверджує, що перше тлумачення стосується якісних властивостей досліджуваних об'єктів, а друге – формальних їхніх властивостей.

Інформаційні технології – це процеси відбору, накопичення, передавання та опрацювання інформації для створення автоматизованих систем різноманітного призначення. Сучасні автоматизовані системи, зазвичай включають інструментальні засоби і є складними системами. Для проектування інформаційних систем предметних галузей широко використовуються методи аналітичного, імітаційного, інфологічного, об’єктно-орієнтованого моделювання. Метод математичного моделювання, порівняно з іншими методами, має такі переваги, як забезпечення можливості точного опису, проведення за вибраними критеріями оптимізації і дослідження моделей. Математичне моделювання інформаційних технологій і систем полягає у побудові і дослідженні їхніх математичних моделей [10]. Синтез і оптимізація математичних моделей інформаційних технологій і систем може бути виконаним на підставі використання модифікованої алгебри алгоритмів [7].

Етапи синтезу моделей

Проектування інструментальних засобів інформаційних технологій виконується на трьох етапах з використанням методів декомпозиції та алгебри алгоритмів. З метою зменшення складності проектування і забезпечення можливості розпаралелення синтезу складних систем на першому етапі виконується структурна декомпозиція системи інструментальних засобів на підсистеми. Структурною декомпозицією визначаються складові і підпорядкованість складових моделі інструментальних засобів інформаційних технологій. Засобами алгебри алгоритмів створюється модель декомпозиції, яка на підставі властивостей операцій алгебри алгоритмів оптимізується за кількістю підсистем кожного із етапів декомпозиції. На другому етапі засобами алгебри алгоритмів синтезується і оптимізується модель функціонування інструментальних засобів. Третій етап полягає у дослідженні, реалізації та апробації математичної моделі інструментальних засобів. У разі виявлення помилок у модель та її реалізацію вносяться необхідні правки.

Декомпозиція структури автоматів. Зазвичай програмно реалізовувані автомати утворені двома складовими. Передусім це інтерфейс користувача, яким здійснюється зв'язок користувача з комп’ютерною системою і використовуються її можливості. Другою складовою є функціональна частина, яка, на відміну від першої складової, є невидимою, але саме вона описує розв’язання тих задач, які візуалізовані в інтерфейсі користувача інструментальних засобів. Уже на самому початку синтезу інструментальних засобів, за функціональним призначенням, систему (S) декомпоновано на графічну (G) і функціональну (F) підсистеми. Ця декомпозиція засобами алгебри алгоритмів описується двома виразами. Один із них використовує операцію секвентування ($\frown$ – знак операції секвентування)

$$S=\overset{\frown}{G,\ F,}$$

яка виражає необхідну наявність обох підсистем і використовує знак коми для розділення, що свідчить про відсутність їхнього впорядкування. За потреби відображення послідовності підсистем використовується яко розділювач “крапка з комою”. У загальному випадку можна було б задати розділювач “дві крапки”, який є ідентифікатором коми і крапки з комою. Друга модель декомпозиції найвищого рівня записується з використанням операції паралелення ($\sqcap$ – знак операції паралелення)

$$S=\overline{G,\ F,}$$

яка виражає незалежну наявність обох підсистем.

Кожна із підсистем, за вибраними критеріями (наприклад, функціональним призначенням), своєю чергою, може теж бути декомпонована на субсистеми. Враховуючи задачі, потреби і можливості, кількість рівнів декомпозиції визначає розробник інструментальних засобів. Формула (1) ілюструє можливість алгебри алгоритмів описати багаторівневу декомпозицію систем. У цій формулі підсистему-унітерм F декомпоновано на складові (підунітерми), які позначимо як P0 , P1 , … , Pp-1. Очевидно, що в F входять усі p унітермів від P0 до Pp-1. Між F і унітермами, на які він декомпонований та описаний засобами алгебри алгоритмів, ставиться знак рівності.

По декомпозиції поточного рівня виконуємо декомпозицію чергового рівня, яка полягає у розбитті унітермів P0 , P1 , … , Pp-1 на підунітерми. Наприклад, унітерм P0 декомпоновано на унітерми, які позначимо як Q0, Q1, … , Qq-1, а унітерм Pp-1 – на унітерми R0, R1, … , Rr-1. Аналогічно розбиваємо всі решта унітермів. Кожному із унітермів P0 , P1 , … , Pp-1 приписуємо паралелення його ”підунітермів” з відповідними розділювачами. Серед унітермів P0 , P1 , … , Pp-1 можливі такі унітерми, декомпозицію яких недоцільно виконувати. Q0, Q1, … , Qq-1 – унітерми першого рівня декомпозиції унітерму P0, а q – кількість унітермів; S0, S1, … , Ss-1 – унітерми передостаннього рівня декомпозиції унітерму Q0, а s – кількість унітермів; U0, U1, … ,Uu-1 – унітерми останнього рівня декомпозиції (функціональні унітерми) унітерму S0, а u – кількість унітермів. Останнім рівнем декомпозиції є декомпонування функціональних унітермів на предметні змінні та оператори-унітерми. Аналогічно описується декомпозиція усіх решта унітермів. Декомпозицією системи інструментальних засобів на підсистеми досягається зменшення складності створення підсистем.

$$F = \left[\begin{array}{l}
P_0 = \left[\begin{array}{l}
Q_0 = \ldots\ \left[\begin{array}{l} S_0 = \overline{U_0; U_1; \ldots U_{u-1}} \\ ; \\ S_1 = \overline{L_0; L_1; \ldots L_{l-1}} \\ \ldots \\ S_{s-1} = \overline{K_0; K_1; \ldots K_{k-1}} \end{array}\right. \\
; \\
Q_1 = \ldots\ \left[\begin{array}{l} E_0 = \overline{A_0; A_1; \ldots A_{a-1}} \\ ; \\ E_1 = \overline{H_0; H_1; \ldots H_{h-1}} \\ \ldots \\ E_{e-1} = \overline{B_0; B_1; \ldots B_{b-1}} \end{array}\right. \\
; \\
\ldots \\
Q_{q-1} = \ldots\ \left[\begin{array}{l} D_0 = \overline{C_0; C_1; \ldots C_{c-1}} \\ ; \\ D_1 = \overline{G_0; G_1; \ldots G_{g-1}} \\ \ldots \\ D_{d-1} = \overline{V_0; V_1; \ldots V_{v-1}} \end{array}\right.
\end{array}\right. \\
\ldots \\
; \\
P_{p-1} = \left[\begin{array}{l}
R_0 = \ldots\ \left[\begin{array}{l} O_0 = \overline{\Delta_0; \Delta_1; \ldots \Delta_{\delta-1}} \\ ; \\ O_1 = \overline{T_0; T_1; \ldots T_{t-1}} \\ \ldots \\ O_{o-1} = \overline{F_0; F_1; \ldots F_{f-1}} \end{array}\right. \\
; \\
R_1 = \ldots\ \left[\begin{array}{l} M_0 = \overline{I_0; I_1; \ldots I_{i-1}} \\ ; \\ M_1 = \overline{J_0; J_1; \ldots J_{j-1}} \\ \ldots \\ M_{m-1} = \overline{W_0; W_1; \ldots W_{w-1}} \end{array}\right. \\
; \\
\ldots \\
R_{r-1} = \ldots\ \left[\begin{array}{l} Z_0 = \overline{N_0; N_1; \ldots N_{n-1}} \\ ; \\ Z_1 = \overline{Y_0; Y_1; \ldots Y_{y-1}} \\ \ldots \\ Z_{z-1} = \overline{X_0; X_1; \ldots X_{x-1}} \end{array}\right.
\end{array}\right.
\end{array}\right. \qquad (1)$$

Декомпозиція функціонування автоматів. Функціональна декомпозиція утворена унітермами-змінними і функціональними унітермами. Інформаційно-технологічні процеси починаються з певної вихідної точки. Нею, наприклад, можуть бути ввімкнення системи чи запуск на виконання. У будь-якому разі має бути виконана певна умова, яку позначимо u, початку функціонування. Якщо умова не виконана, то відбувається очікування її виконання, що запишемо поверненням у цикл (cu) за умовою u. Коли ж умова виконана, то, наприклад, функціонування починається з унітерму Z. У вигляді формули процес очікування виконання умови опишемо формулою [11]

$$\overset{\not\subset u}{\overline{Z;\ c_u;\ u\text{-}?}}\ .$$

Зазвичай, Z задаються початкові значення.

Типові інструментальні засоби мають інтерфейс користувача. Складові інтерфейсу (%Ai, %Bj, %Ck, …) вибираються (ai, bj, ck, … – умови вибору відповідної складової інтерфейсу) користувачем системи інструментальних засобів, що опишемо формулою

$$Y_r = \overline{\overline{\overline{\%A_i; *; a_i\text{-}?}, \overline{\%B_j; *; b_j\text{-}?}, \overline{\%\, C_k; *; c_k\text{-}?}}, \ldots} ,$$

у якій знак % є ідентифікатором складової інтерфейсу, а знаком операції паралелення з розділювачем унітермів комою описано можливість вибору будь-якої графічної складової.

Кожна типова графічна складова %Ai, %Bj, %Ck, … утворена своїми властивостями (Wi, Wi, Wk) і назвою функціонального унітерму, який запускається на виконання у разі вибору графічної складової. Наприклад,

$$\%A_i = \overset{\frown}{W_i, A_i}.$$

Враховуючи все вищеописане, загальна формула функціональної декомпозиції опишеться таким виразом:

$$\overline{\left(\begin{array}{l} Z;\ c_u;\ u\text{-}? \\ \quad : \\ \overline{\%A_i;\ \overline{\%B_j;\ \overline{\%C_k;\ \ldots;\ c_k\text{-}?};\ b_j\text{-}?}\ ;\ a_i\text{-}?} \end{array}\right.}^{\,\not\subset u}$$

Отож алгеброю алгоритмів описуються інтерфейси користувачів і функціонування систем інструментальних засобів інформаційних технологій.

Оптимізація математичних моделей автоматів. Можливості, які забезпечує алгебра алгоритмів для оптимізації моделей автоматів, проілюструємо таким твердженням:

Теорема. З формули алгоритму

$$\overline{\left(\begin{array}{l} A \\ ; \\ X \\ ; \\ B \end{array}\right.;\ \left(\begin{array}{l} C \\ ; \\ X \\ ; \\ D \end{array}\right.;\ u\text{-}?} \qquad (2)$$

введенням додаткової змінної p з приписуванням їй значень 0 і 1, від якої не залежать унітерми формули, виводиться вираз

$$\left(\begin{array}{l} \overline{\left(\begin{array}{l} A\ ; \\ , \\ p{=}0 \end{array}\right. \left(\begin{array}{l} C;\ u\text{-}? \\ , \\ p{=}1 \end{array}\right.} \\ ; \\ \left(\begin{array}{l} X \\ ; \\ \overline{B;\ D;\ (p{=}0)\text{-}?} \end{array}\right. \end{array}\right. \qquad (2а)$$

з меншою кількістю унітермів X.

Доведення. Якщо умова u виконується, то на основі означення операції елімінування [5, 6] виконуємо такі перетворення:

$$\left(\begin{array}{l} \overline{\left(\begin{array}{l} A\ ; \\ , \\ p{=}0 \end{array}\right. \left(\begin{array}{l} C;\ u\text{-}? \\ , \\ p{=}1 \end{array}\right.} \\ ; \\ \left(\begin{array}{l} X \\ ; \\ \overline{B;\ D;\ (p{=}0)\text{-}?} \end{array}\right. \end{array}\right. = \left(\begin{array}{l} \left(\begin{array}{l} A \\ , \\ p{=}0 \end{array}\right. \\ ; \\ \left(\begin{array}{l} X \\ ; \\ \overline{B;\ D;\ (p{=}0)\text{-}?} \end{array}\right. \end{array}\right. = \left(\begin{array}{l} \left(\begin{array}{l} A \\ , \\ p{=}0 \end{array}\right. \\ ; \\ \left(\begin{array}{l} X \\ ; \\ B. \end{array}\right. \end{array}\right. \qquad (3)$$

Друга рівність має унітерм B, який отриманий із елімінування попередньої рівності за умовою (p=0)-?, що виконується.

Якщо ж умова u не виконується, то отримуємо аналогічні рівності

$$\left(\begin{array}{l} \overline{\left(\begin{array}{l}A\\ ,\\ p=0\end{array}\right) ; \left(\begin{array}{l}C\\ ,\\ p=1\end{array}\right) ; u\text{-?}} \\ ; \\ \left(\begin{array}{l} X \\ ; \\ \overline{B;\ D;\ (p=0)\text{-?}} \end{array}\right) \end{array}\right) = \left(\begin{array}{l} \left(\begin{array}{l}C\\ ,\\ p=1\end{array}\right) \\ ; \\ \left(\begin{array}{l} X \\ ; \\ \overline{B;\ D;\ (p=0)\text{-?}} \end{array}\right) \end{array}\right) = \left(\begin{array}{l} \left(\begin{array}{l}C\\ ,\\ p=1\end{array}\right) \\ ; \\ \left(\begin{array}{l} X \\ ; \\ D. \end{array}\right) \end{array}\right) \quad (4)$$

У такому разі умова (p=0)-? елімінування передостанньої рівності не виконується, тому з цього елімінування отримується унітерм D.

З формули (2) у разі виконання умови u на підставі означення операції елімінування [4, 5] виводимо вираз

$$\left(\begin{array}{l} A \\ ; \\ \left(\begin{array}{l} X \\ ; \\ B. \end{array}\right) \end{array}\right) \quad (5)$$

Коли ж умова u не виконується, то аналогічно отримуємо вираз

$$\left(\begin{array}{l} C \\ ; \\ \left(\begin{array}{l} X \\ ; \\ D. \end{array}\right) \end{array}\right) \quad (6)$$

У виразі (3) значення змінної p не впливає на значення усіх решта її унітермів (A, X і B), тому з точністю до значень унітермів формули (3) і (5) є однакові. Аналогічно є однаковими формули (4) і (6). Отже, введення змінної p, яка не змінює значень унітермів формули (2), не змінює значення алгоритму. Теорему доведено.

Як це проілюстровано теоремою, введенням додаткової змінної p формула (2) оптимізується до формули (2a), у якій вже відсутнє дублювання змінної X. В алгоритмах інструментальних засобів та інформаційних технологій замість змінної X можуть бути десятки і більше функціональних унітермів. Застосування алгебри алгоритмів забезпечує виконання оптимізації алгоритмів без зміни їхньої функціональності.

Дослідження математичних моделей автоматів. Метод дослідження математичних моделей автоматів базується на застосуванні повної і трансфінітної математичної індукції. Для виконання дослідження необхідним є задання секвентних областей значень (впорядкованих множин значень) усіх унітермів-змінних і функціональних унітермів. Доконечним є врахування впорядкованості унітермів-змінних і функціональних унітермів. Дослідження виконується для початкових, кінцевих та одного із будь-яких проміжних значень усіх унітермів-змінних і функціональних унітермів. Виконаними дослідженнями алгоритмів установлюється забезпеченість ними очікуваних результатів для усіх заданих значень змінних і функціональних унітермів.

Висновки

1. Методологія синтезу алгоритмічного забезпечення автоматів передбачає використання методів декомпозиції, алгебри алгоритмів і повної та трансфінітної математичної індукції.
2. Алгебра алгоритмів використовується на етапах структурного і функціонального синтезу, оптимізації і дослідження математичних моделей автоматів.
3. Засобами одноосновної алгебри алгоритмів описуються математичні моделі автоматів.

1. Драґан Я.П., Медиковський М.О., Овсяк В.К., Сікора Л.С., Яворський Б.І. Системний аналіз проблем управління та кібернетики як підстави інформаційних технологій // Праці конференції

Обчислювальні методи і системи перетворення інформації, 4–5 жовтня, 2012. 2. Автоматы. Сб. статей / Под ред. К.Э. Шенона, Дж. Маккарти, перев. с англ. под ред. А.А. Ляпунова. – М: ИИЛ, 1956. – 604 с. 3. Математика. Большой энциклопедический словарь / Гл. ред. Ю.В. Прохоров, 3-е изд. – М.: Большая Рос. энциклопедия, 1998. – 848 с. 4. Глушков В.М., Цейтлин Г.Е., Ющенко Е.Л. Алгебра. Языки. Программирование. – 2-е изд. перераб. – Киев: «Наукова думка», 1978. – 318 с. 5. Цейтлин Г.Е. Введение в алгоритмику. – К.: Сфера, 1998. – 310 с. 6. Овсяк В.К. Засоби еквівалентних перетворень алгоритмів інформаційно-технологічних систем / Овсяк В.К. // Доповіді Національної академії наук України. – 1996. – № 9. – С. 83–89. 7. Owsiak W., Owsiak A. Rozszrzerzenie algebry algorytmόw //Pomiary, automatyka, kontrola. – 2010. – № 2. – S. 184–188. 8. Математическая энциклопедия. – Т.4. – М.: Советская энциклопедия, 1982. – 1216 стб. 9. Драган Я., Овсяк В., Сікора Л. Самоорганізація – принцип еволюції складних систем // Дух, наука, думка, воля поступу української економіки: Зб. праць. / Упоряд. І.В. Барановський. – Львів: Укр. акад. друк. 2001. – 64 с. – С. 29–31. 10.Математическая энциклопедия. – Т. 3. – М.: Советская энциклопедия, 1982. – 1184 стб. 11. Овсяк О.В. Рекурентно-декомпозиційна методологія інформаційних технологій і систем // Поліграфія і видавнича справа. – 2011. – № 3 (55). – С. 74–84.

УДК 621.372:538.56

В. Передерій[1], С. Бабичев[2], В. Литвиненко[1, 2]
[1]Миколаївський національний університет ім. О.М. Сухомлинського,
кафедра комп'ютерних систем та мереж
[2]Херсонський національний технічний університет,
кафедра інформаційних технологій та комп'ютерних систем

ЗАСТОСУВАННЯ МЕРЕЖІ БАЙЄСА ДЛЯ ОЦІНКИ СТУПЕНЯ ЗНАЧИМОСТІ ВПЛИВАЮЧИХ ФАКТОРІВ НА ЛПР В АВТОМАТИЗОВАНИХ СИСТЕМАХ ПРИ ПРИЙНЯТТІ РЕЛЕВАНТНИХ РІШЕНЬ

Розроблено Байєсівську мережу для оцінки ступеня значимості особистих і зовнішніх факторів, що впливають на прийняття релевантних рішень ЛПР в автоматизованих системах управління.

Ключові слова: особа що приймає рішення, прийняття релевантний рішень, особистісні і зовнішні фактори, Байєсовська мережа.

Developed and investigated Bayesian network designed to assess the significance of personal and external factors affecting the performance of the automated control system

Key words: individual decision-maker, decision-making relevant, personal and external factors, Bayesian network, the coefficients for the change in the probability factor

Вступ

Сучасні автоматизовані системи прийняття рішень характеризуються наявністю складного об'єкта управління з розподіленими параметрами, неоднорідних і інтенсивних інформаційних потоків, що надходять до користувача в режимі реального часу. Робота користувача системи під час управління технологічними об'єктами є напруженою, а наслідки помилок призводять до значного матеріального збитку, людських жертв тощо [1]. Внаслідок цього високу актуальність набуває