УДК 004.416.3; 004.855.5

П.І. ФЕДОРУК, М.В. ПІКУЛЯК

ВИКОРИСТАННЯ МАТЕМАТИЧНОЇ ЛОГІКИ ДЛЯ ВСТАНОВЛЕННЯ НАВЧАЛЬНОГО РЕЖИМУ В АВТОМАТИЗОВАНИХ СИСТЕМАХ ПЕРЕДАЧІ ЗНАНЬ

Анотація. *Проведено аналіз використання апарату математичної логіки в задачах програмування та досліджено ефективність застосування тризначної математичної логіки для обробки сценарних прикладів з метою побудови адаптивної навчальної траєкторії в автоматизованих системах передачі знань.*
Ключові слова: *адаптивна система, сценарні приклади, математична логіка.*

Аннотация. *Проведен анализ использования аппарата математической логики в задачах программирования и исследована эффективность применения трёхзначной математической логики для обработки сценарных примеров с целью построения адаптивной учебной траектории в автоматизированных системах передачи знаний.*
Ключевые слова: *адаптивная система, сценарные примеры, математическая логика.*

Abstract. *Usage of apparatus of mathematical logic in the programming problems was analyzed and the effectiveness of three-digit mathematical logic for scenario examples processing with the purpose of building adaptive learning trajectory in automated systems of knowledge transfer was investigated.*
Keywords: *adaptive system, script examples, mathematical logic.*

1. Вступ

Сучасний стан освіти в Україні та тенденції розвитку інформаційного суспільства визначають необхідність розробки якісно нових навчально-методичних засобів, орієнтованих на застосування новітніх інформаційних технологій з метою побудови навчальних автоматизованих систем.

Беручи до уваги вимоги педагогіки, найбільш затребуваними в навчальному процесі виступають саме адаптивні системи, що забезпечують реалізацію програмним методом індивідуального підходу до навчання, враховуючи при цьому початковий рівень знань студента та структуру навчального матеріалу.

Аналіз сучасних адаптивних Web-систем дозволяє розділити їх на три групи [1]:

– адаптивні інформаційні системи, що служать для персоналізації інформації в режимі on-line (AVANTI, PUSH);

– адаптивні фільтруючі системи, які допомагають користувачеві знаходити релевантні "перегляди" в океані доступної інформації (ifWeb, WebTagger & trade, AHA, WebCOBALT);

– навчальні адаптивні системи (FLINT, MONAP-II, ELM-ART, CALAT, WITS, Belvedere та ін.).

Дані системи розроблені на основі методів системного аналізу, теорій множин та автоматів, теорії та методів програмної інженерії, методів об'єктно-орієнтованого аналізу та проектування, методів візуального моделювання. Для реалізації механізмів адаптації використовуються експертні системи, дискретні математики, марковські процеси, мережі Петрі, а також об’єктні моделі обробки навчально-методичної інформації.

Незважаючи на значну чисельність розроблених принципів та методів адаптації, програмна реалізація самої техніки адаптації залишається унікальною для кожної навчальної системи. Адже автору-розробнику, який створює нову чи доповнює уже розроблену систему навіть уже відомим принципом адаптації, необхідно заново будувати модель користувача, алгоритми, програмні рішення, форми представлення даних з метою реалізації навчального ефекту для своєї системи з унікальною архітектурою.

Все це свідчить про актуальність проблеми подальших досліджень розробки новітніх методів та програмних засобів для побудови адаптивних систем, використовуючи передові технології структурування та метаопису даних.

2. Постановка задачі дослідження

Метою даної роботи є вирішення задачі застосування апарату математичної логіки, а саме досконалої диз'юнктивної нормальної форми (ДДНФ) та її властивостей для дослідження адаптації автоматизованої системи до навчальної поведінки студента з метою пошуку сценарного правила, продовження навчання, яке найбільш точно відповідає поточному рівню знань та індивідуальним особливостям того, хто навчається.

Це включає також аналіз існуючих методів і технологій у програмуванні, які використовують апарат математичної логіки та побудову структури сценарних прикладів у моделі навчальної системи.

3. Порівняльний аналіз застосування засобів математичної логіки в задачах програмування

Перші спроби застосувати у програмуванні логічні обчислення та методи формалізації зробив американський логік Х.Б. Каррі, який розглядав задачу програмування як співставлення більш великих програм з окремих готових компонентів [3]. Ним було введено дві базисні системи конструкцій: перша – послідовне виконання, розгалуження і цикл, друга – послідовне виконання та умовний перехід.

У зв'язку з тим, що в 60-х роках на перший план вийшли задачі точного визначення формальних мов досить складної структури, засоби математичної логіки було успішно застосовано для опису синтаксису мов програмування.

Зокрема, в 1965 р. академік В.М.Глушков ввів поняття алгоритмічної алгебри, що послужило прообразом алгоритмічних логік. Швейцарський математик Ф.Енгелер у 1967 р. запропонував використовувати мови з нескінченними формулами, щоб дослідити нескінченну множину можливостей (варіантів), що виникають при різних виконаннях програми.

Проте найбільшої популярності набули мови алгоритмічних логік, які були винайдені практично одночасно американськими логіками Р.У. Флойдом (1967), С.А.Р. Хоаром (1969) та вченими польської логічної школи (А.Сальвіцкий та ін. (1970)) [3]. Дані мови ґрунтуються на логіці предикатів 1-го порядку і містять у собі висловлювання вигляду {A} S {B}, що трактуються таким чином: «Якщо до виконання оператора S було виконано A, то після нього буде виконано B ». Тут A називається передумовою, B – післяумовою S. На цій мові даються логічні описи операторів присвоєння, умовного переходу, розгалуження та циклу.

Принципово інший спосіб визначення семантики програм, характерний більше для опису всієї алгоритмічної мови, а не окремих програм, запропонував у 1970 р. американський логік Д. Скотт [4]. Він побудував математичну модель λ-обчислення і показав, як переводити функціональний опис мови структурного програмування в λ-обчислення і яким чином на її основі визначити математичну модель алгоритмічної мови. Ця так звана денотаційна семантика алгоритмічних мов стала практичним інструментом побудови сучасних трансляторів зі складних алгоритмічних мов. Структурні схеми відповідали новому типу логіки – логіці схем програм, яку використовує програміст для створення складних, багатоваріантних, ітеративних планів дій.

Вперше застосовувати математичну логіку як інструмент аналізу понять програмування почали з середини 1970-х років. Тоді було доведено, що для багатьох конструкцій програмування доцільно використовувати окремі логічні комбінації, а не їх сукупність, що часто приводить до неможливості проведення алгоритмічних обчислень. Наприклад, оператори GO-TO природно застосовувати в тих випадках, коли дії, що розглядаються, можна вважати глобальними перетвореннями стану системи. Цикли виявились добре сумісними із масивами та погано – з рекурсивними структурами даних, а процедури вищих типів – навпаки. Масиви і складні структури даних погано сумісні з присвоюванням [5].

Проведений огляд доводить, що застосування апарату математичної логіки залежить від визначеного класу задач, що досліджуються, та структури об’єктів, над якими проводять алгоритмічні обчислення. Його вибір для обробки сценарних правил в адаптивній навчальній системі обумовлений можливістю використання потужного математичного інструменту для виконання розрахунків та проведення подальшого аналізу ефективності застосування розробленого інструментального засобу на практиці.

4. Структурна модель сценарного прикладу

У розробленій автоматизованій системі напрямок навчальної траєкторії для окремого студента визначає домен-експерт на основі аналізу поточних результатів засвоєння ним матеріалу.

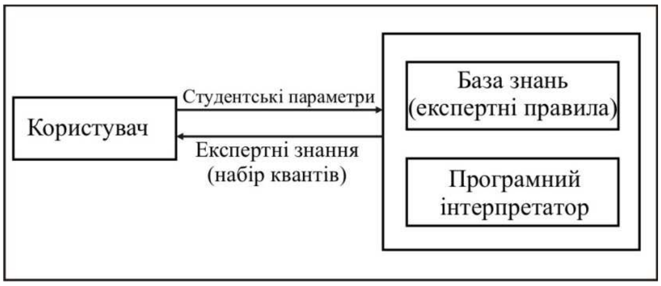

Рис. 1. Схема процесу функціонування експертної системи

Відповідно до результатів проходження контрольного тестування для кожного, хто навчається, обчислюються значення параметрів студентської моделі P_i. Дані параметри передаються в експертну систему, на основі яких, з використанням відповідних предикатних правил, програмний інтерпретатор формує відповіді системи у вигляді експертних знань. Такі знання представляють набір квантів, необхідних для продовження навчального процесу (рис. 1) [6].

У результаті проведеного аналізу розроблених навчальних програм і моделей студента для оцінки поточного рівня засвоєння знань було обрано такі параметри, значення яких отримуємо із студентського модуля [7]:

P_1 – числове значення, яке визначає загальний рівень засвоєння навчального матеріалу;

P_2 – глибина знань;

P_3 – рівень засвоєння з окремих розділів (блоків);

P_4 – швидкість засвоєння;

P_5 – швидкість сприйняття.

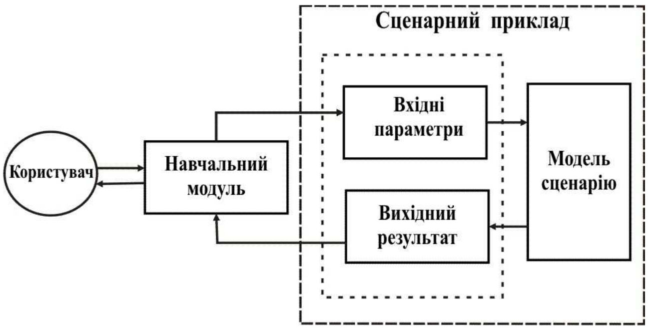

Рис. 2. Представлення сценарного прикладу в моделі навчальної системи

Дані параметри використовуються для побудови бази сценарних прикладів, кожен з яких являє собою окреме педагогічне рішення, подане у вигляді математичної моделі, що відображає відношення між навчальними квантами (рис. 2).

Взагалі під час сценарного дослідження вибудовується гіпотетична картина послідовного розвитку в часі навчальної поведінки студента, яка в сукупності складає еволюцію засвоєння навчального матеріалу.

Оскільки сценарні приклади будуються в рамках припущень про можливий напрямок продовження навчання, то вони являють собою деяку відносну, умовну оцінку можливої поведінки навчальної системи. Іншими словами, кожен сценарний приклад намагається дати відповідь на запитання: «Якою буде поведінка системи, якщо вхідні студентські параметри будуть приймати саме такі значення?».

Згідно з загальною структурою побудови сценарного прикладу, під час перевірки виконання чи невиконання умови відбувається перенаправлення навчальної траєкторії на певний режим продовження навчання. Кількість можливих альтернатив сценарного розвитку, тобто подальших напрямків навчальної траєкторії, визначається особливістю параметрів, що досліджуються в системі. У випадках, коли параметри, що позитивно впливають на рівень засвоєння контенту, набувають максимально можливих значень, а ті параметри, значення яких негативно характеризують навчальний рівень студента, мають мінімально можливі значення – навчальна траєкторія студента, з точки зору рівня засвоєння матеріалу, набуває найбільш оптимального напрямку розвитку. Це свого роду верхня межа можливого успішного розвитку навчального процесу. Мінімальну межу визначають навпаки: мінімізують позитивні параметри і максималізують негативні. Студентські параметри, що знаходяться між максимальними і мінімальними значеннями, визначають інші можливі альтернативи продовження навчання із залученням відповідних навчальних квантів.

Широкий спектр різних варіантів сценарного продовження навчання дозволяє розробити гнучку систему дій системи на можливі значення досліджуваних студентських параметрів. Це дає змогу вирішити основну задачу сценарію – максимально зменшити ступінь невизначеності в системі.

Наведемо зразок побудови сценарного прикладу [8],

Таблиця 1. Сценарний приклад

ЯКЩО				ТО
P_1	P_2	$\dots$	P_i	$R_j,\ j=1,2,3$
С	Н		Н	$R_1 \to 0{,}1$
…				…
С	В		С	$R_2 \to 0{,}4$
…				…
В	С		В	$R_3 \to 0{,}9$

де $P_1, P_2, \dots, P_i$ – параметри оцінки засвоєння знань студентом;

R_1, R_2, R_3 – відповідно режим перенавчання, донавчання та навчання;

В; С; Н – відповідно «високе», «середнє», «низьке» значення параметрів P_i.

Кожен рядок табл. 1 представляє взаємозв’язки між можливими значеннями параметрів P_i та значеннями імплікації R_j, що відповідає їм. Таке представлення дає можливість відобразити поточний рівень засвоєних знань студентом у вигляді матриці, клітинки якої моделюють факти та навички, здобуті ним під час навчання.

Наведений сценарний приклад доцільно представити у формі термінального кванта 2-го рівня [9], що спрощує подальше його опрацювання за допомогою формул математичної логіки:

$$vk_2R_j = \begin{bmatrix} P_1 \big| C : P_2 \big| H :\dots: P_i \big| H : R_1(\to) = 0{,}1 \\ \dots \\ P_1 \big| C : P_2 \big| B :\dots: P_i \big| C : R_2(\to) = 0{,}4 \\ \dots \\ P_1 \big| B : P_2 \big| C :\dots: P_i \big| B : R_3(\to) = 0{,}9 \end{bmatrix}. \qquad (1)$$

У розробленій системі поточний рівень засвоєних знань студентом формується студентським модулем у вигляді вектора $P(P_1, P_2, ...P_k)$, що передається в домен-експерт для пошуку подальшого режиму продовження навчального процесу. З цією метою застосовується математичний апарат, побудований на законах математичної логіки. Беручи до уваги те, що кожен із параметрів P_i може набувати трьох значень (В,С,Н), будується досконала диз'юнктивна нормальна форма [10] для тризначної математичної логіки. Це дає можливість, використовуючи правила множення елементарних кон'юнкцій, математично визначити номер режиму R_j, що перенаправляє навчальну траєкторію на повторне чи поглиблене вивчення деякої частини навчального контенту (рис. 3).

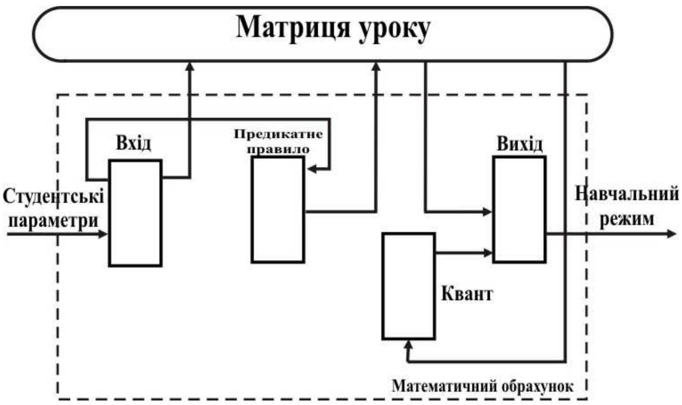

Рис. 3. Структура сценарного прикладу

Такий згенерований пакет квантів подається на вивчення у вигляді відповідного інформаційного потоку. Базове ядро навчального контенту визначається конкретним змістом курсу (теми), що вивчається. Це можуть бути, наприклад, елементи теорії або методичні рекомендації для виконання лабораторного практикуму чи вправ, перелік задач та завдань для повторного вивчення та інші види знань. Таким чином, для кожного студента формується та видається індивідуальне методично обґрунтоване завдання. В заключній частині кожного завдання видається серія контрольних запитань (тест). Відповіді, отримані системою, змінюють поточні значення вектора стану окремого студента і у вигляді інформаційного потоку знову передаються в домен-експерт.

5. Застосування тризначної логіки для математичної обробки матриці уроку

Для побудови тризначної математичної логіки, яка використовується в адаптивній системі для визначення навчального режиму, скористаємось тризначною логікою Я. Лукасевича [11]. Як пропозиційні змінні виберемо описані вище змінні В,С,Н.

Нехай М – непорожня множина $\{B, C, H\}$, де змінні В,С,Н інтерпретуються як логічні значення, а саме високе, середнє та низьке значення терм (студентських параметрів), що визначені на певній області значень з проміжку [0..1]:

В («високий») – [0,8..1];

С («середній») – [0,4..0,8);

Н («низький») – [0..0,4).

Формули логіки висловлювань (пропозиційні формули) будемо будувати за допомогою введених пропозиційних змінних та формальних символів – дужок і знаків, що позначають операції над висловлюваннями:

$\wedge$ – кон'юнкція, операція «І»;

$\vee$ – диз'юнкція, операція «Або»;

$\rightarrow$ – імплікація, операція «Якщо…, то».

Пропозиційні формули будуються, починаючи з пропозиційних змінних за допомогою такого правила [9]:

"Якщо А і В є формулами логіки висловлювань, то можна побудувати нові формули: $(A \wedge B)$, $(A \vee B)$, $(A \rightarrow B)$".

Означення: тризначним предикатом Р розмірності n на множині М будемо називати будь-яке відображення M^n на множині $\{B, C, H\}$.

Тобто, тризначний предикат P на множині M буде деяка n-містна функція, визначена на M із значеннями на множині $\{B, C, H\}$.

Оскільки у розробленій системі рівень оволодіння студентом навчальним матеріалом студентський модуль визначає на основі аналізу 5-ти параметрів P_i, то для 5-містної функції предикат P буде позначений як

$$P(C_1, H_2, C_3, B_4, H_5), Q(B_1, C_2, B_3, H_4, C_5), R(H_1, B_2, C_3, C_4, B_5) \text{ і т.д.}$$

У даному випадку квант (1) буде мати такий вигляд:

$$vk_2R_j = \begin{bmatrix} P_1 \big| C : P_2 \big| H : P_3 \big| B : P_4 \big| B : P_5 \big| C : R_1(\rightarrow) = 0,1 \\ ... \\ P_1 \big| C : P_2 \big| C : P_3 \big| B : P_4 \big| H : P_5 \big| B : R_2(\rightarrow) = 0,4 \\ ... \\ P_1 \big| B : P_2 \big| B : P_3 \big| H : P_4 \big| C : P_5 \big| B : R_3(\rightarrow) = 0,1 \end{bmatrix}. \tag{2}$$

З метою спрощення подальших обчислень для першого рядка кванту (2) введемо такі позначення:

$$\begin{array}{cccccc} P_1 \,|\, C & P_2 \,|\, H & P_3 \,|\, B & P_4 \,|\, B & P_5 \,|\, C & R_1(\rightarrow) = 0.1 \\ \downarrow & \downarrow & \downarrow & \downarrow & \downarrow & \downarrow \\ C_1 & H_2 & B_3 & B_4 & C_5 & R_{11} \end{array},$$

де B,C,H – значення терм, нижній індекс відображає прив’язку терми до відповідного параметра P_i; R_{11} – номер навчального режиму.

Аналогічно вводяться позначення для всіх інших рядків (2). Тоді квант (2) можемо представити такою матрицею:

$$\begin{bmatrix} C_1 \wedge H_2 \wedge B_3 \wedge B_4 \wedge C_5 \,|\, R_{11} \\ ... \\ C_1 \wedge C_2 \wedge B_3 \wedge H_4 \wedge B_5 \,|\, R_{24} \\ ... \\ B_1 \wedge B_2 \wedge H_3 \wedge C_4 \wedge B_5 \,|\, R_{31} \end{bmatrix}, \tag{3}$$

кожен рядок якої буде являти собою елементарну кон’юнкцію з імплікацією на відповідний режим навчання R_{ij}.

Для проведення математичного опрацювання визначеної тризначної логіки побудуємо для матриці (3) аналог досконалої диз’юнктивної нормальної форми (ДДНФ).

Використовуючи відомі закони логіки, будь-яку формулу можна перетворити у рівносильну їй формулу вигляду $C_1 \vee C_2 \vee ... C_m$, де $m \geq 1$ і кожне C_i або змінна, або її заперечення, або кон’юнкція змінних чи їх заперечень. Формула такого типу називається диз’юнктивною нормальною формою (ДНФ) даної формули [11].

Оскільки кожен рядок (3) можна розглядати як кон’юнкцію елементарних висловлювань, що відповідають різним наборам значень P_i, то з’єднавши всі кон’юнкції знаком диз’юнкції, можна представити матрицю (3) у формі ДДНФ [12]:

$$C_1C_2B_3B_4C_5 \,|\, R_{11} \;\; \vee \;\; ... \;\; \vee \;\; C_1C_2B_3H_4B_5 \,|\, R_{24} \;\; \vee \;\; ... \;\; \vee \;\; B_1B_2H_3C_4B_5 \,|\, R_{31}\,. \tag{4}$$

Будемо вважати, що за результатами поточного тестування рівень оволодіння навчальним матеріалом для деякого студента М представлений, наприклад, вектором $P(C_1, C_2, B_3, H_4, B_5)$. Використовуючи ДДНФ, визначимо номер режиму, за яким необхідно продовжити навчання для студента М.

Беручи до уваги правила множення у двозначній логіці [10]:

$$|P \wedge P| = 1$$
$$|P \wedge \overline{P}| = 0,$$

введемо аналогічні кон'юнктивні правила для означеної вище тризначної логіки, а саме:

$$|C_i \wedge C_j| = 1, \textit{якщо} \quad i = j,$$
$$|C_i \wedge C_j| = 0, \textit{якщо} \quad i \neq j,$$
$$|C_i \wedge B_j| = 0, \textit{при} \quad \textit{довільних} \quad \textit{значеннях} \quad i, j.$$

Розглядаючи вектор P як один кон'юнктивний член ДДНФ та використовуючи означені вище кон'юнктивні правила, в результаті множення вектора P на матрицю уроку, представлену у формі (4)

$$C_1C_2B_3H_4B_5(C_1C_2B_3B_4C_5 | R_{11} \quad \vee \quad ... \quad \vee \quad C_1C_2B_3H_4B_5 | R_{24} \quad \vee \quad ... \quad \vee \quad B_1B_2H_3C_4B_5 | R_{31}) = 11111 | R_{24},$$

отримаємо шуканий номер режиму R_{24}, продовження навчання за яким дає змогу адаптувати навчальний контент відносно успіхів, які досягнув студент М на деякому поточному етапі навчання.

6. Висновки

У роботі розроблено математичний механізм обробки сценарних прикладів з метою побудови адаптивної навчальної траєкторії в автоматизованих системах передачі знань.

Програмна реалізація описаної технології дає змогу вирішити такі навчальні задачі:

• визначити номер навчального режиму під час адаптивного навчання;

• мінімізувати витрачений час та кількість навчального контенту;

• адаптувати навчальний контент відповідно до здобутих успіхів студента під час навчання;

• забезпечити індивідуальний підхід до кожного студента шляхом подання на вивчення навчального матеріалу за наперед розробленими сценарними прикладами;

• математично обгрунтувати правильність проведених розрахунків.

СПИСОК ЛІТЕРАТУРИ

1. Брусиловский П.Л. Адаптивные обучающие системы в World Wide Web: обзор имеющихся в распоряжении технологий [Электронный ресурс] / П.Л. Брусиловский // Режим доступа: http://ifets.ieee.org/russian/depository/WWWITS.html.

2. Карри Х.Б. Основания математической логики / Карри Х.Б.; пер с англ. – М.: Мир, 1969. – 568 с.

3. Гуц А.К. Математическая логика и теория алгоритмов: [учеб. пособие для вузов] / Гуц А.К. – Омск: Наследие. Диалог-Сибирь, 2003. – 108 с.

4. Scott D. Models for the λ-calculus [Рукопись (неопубл.)] / Scott D. – 1969. – 53 p.

5. Игошин В.И. Математическая логика и теория алгоритмов / Игошин В.И. – [2-е изд., стер.]. – М.: Академия, 2008 – 448 с.

6. Марценюк В.П. Побудова бази знань в адаптивній навчальній системі / В.П. Марценюк, П.І. Федорук, М.В. Пікуляк // Вісник Київського університету. – (Серія «Фізико-математичні науки»). – Київ, 2011. – № 3. – С. 193 – 199.

7. Федорук П.І. Адаптивна передача знань із використанням багатопараметричної моделі студента / П.І. Федорук, С.М. Масловський // Науковий вісник Чернівецького національного університету. – (Серія «Комп'ютерні системи та компоненти»). – Чернівці: ЧНУ, 2011. – Т. 2, Вип. 2. – С. 91 – 96.
8. Федорук П.И. Использование сценарных примеров знаний при построении индивидуальной учебной траектории / П.И. Федорук, Н.В. Пикуляк // Программные продукты и системы. – 2011. – № 2 (94). – С. 89 – 94.
9. Сироджа И.Б. Квантовые модели и методы искусственного интеллекта для принятия решений и управления / Сироджа И.Б. – К.: Наукова думка, 2002. – 427 с.
10. Колмогоров А.Н. Математическая логика / А.Н. Колмогоров, А.Г. Драгалин. – [3-е изд., стер.]. – М.: КомКнига, 2006. – 240 с.
11. Lukasiewicz J. O logice trojwartosciowej / J. Lukasiewicz // Ruch hlozohczny. – 1920. – N 5. – P. 168 – 171.
12. Никольская И.Л. Математическая логика: учебник / Никольская И.Л. – М.: Высшая школа, 1981. – 127 с.

Стаття надійшла до редакції 26.04.2012