DOI https://doi.org/10.15589/znp2020.3(481).12
УДК 001.891:65.011.56

OPTIMIZATION OF A REGULAR LOGICAL TREE BY PERMUTATION OF STRUCTURAL ELEMENTS

ПИТАННЯ ОПТИМІЗАЦІЇ ЛОГІЧНОГО ДЕРЕВА ШЛЯХОМ ПЕРЕСТАНОВКИ СТРУКТУРНИХ ЕЛЕМЕНТІВ

Igor F. Povkhan
igor.povkhan@uzhnu.edu.ua
ORCID: 0000-0002-1681-3466

І. Ф. Повхан,
канд. техн. наук, доцент

Uzhgorod National University, Uzhgorod
Ужгородський національний університет, м. Ужгород

Abstract. The paper continues the general problem of structures (methods and schemes) of logical classification trees and raises important questions related to the general methodology for optimizing (minimizing and pruning) tree-like logical structures (LCT/ACT models) by rearranging structural elements in their design. A simple, efficient and economical structure of the logical classification tree (LCT/ACT models) with a minimum sparsity coefficient (based on the initial training sample) allows you to provide the necessary speed, the level of complexity of the recognition scheme, which guarantees simple and complete recognition of discrete objects. Representing the initial training sample in the form of a logical tree generates a tree-like data structure that provides compression and transformation of the initial data of the training sample, and therefore allows significant optimization and saving of hardware resources of the information system. An arbitrary classification tree can be written as a set of logical functions. Then important problems in building recognition systems of this type will be the problems of synthesizing logical functions that are equivalent to a given recognition tree, estimating their complexity, and minimizing the resulting tree. The complexity of graph-schema models (structures of logical classification trees) that are constructed during learning of the recognition system (the logical classification tree is actually a generated recognition function) is investigated. This paper will be devoted to the issue of optimizing the design of logical trees by rearranging structural elements in their design. To date, there are certain approaches to minimizing (normalizing) logical tree structures (LCT/ACT), which are characterized by a certain algorithmic complexity and a rigid focus on specific logical graph – schema representations (classes of logical trees). The method of rearranging tiers (blocks of LCT construction) in the structure of logical trees allows you to achieve a significant effect during optimization and can be applied to a regular Tree of arbitrary complexity. Also, it is necessary to note the significant advantages of this approach of minimizing logical classification trees in terms of software simplicity of building classification trees, reducing the time of overall generation of a logical tree, and so on. This work is relevant for all image recognition methods in which the resulting classification function can be represented as a logical tree.
Key words: permutation of tiers; logical tree; graph-schema models; minimization of logical trees.

Анотація. Робота продовжує загальну проблематику структур (методів та схем) логічних дерев класифікації та порушує важливі питання, які пов'язані із загальною методикою оптимізації (мінімізації й обрізки) деревоподібних логічних конструкцій (моделей ЛДК/АДК) шляхом перестановки структурних елементів в їх конструкції. Проста, ефективна й економна структура логічного дерева класифікації (моделі ЛДК/АДК) із мінімальним коефіцієнтом розрідженості (на основі початкової навчальної вибірки) дозволяє забезпечити необхідну швидкодію, рівень складності схеми розпізнавання, що гарантує проведення простого та повного розпізнавання дискретних об'єктів. Представлення початкової навчальної вибірки у вигляді логічного дерева генерує деревоподібну структуру даних, яка забезпечує стиск та перетворення початкових даних навчальної вибірки, а отже, дозволяє суттєву оптимізацію й економію апаратних ресурсів інформаційної системи. Довільне дерево класифікації можна записати у вигляді набору логічних функцій. Тоді важливими проблемами під час побудови систем розпізнавання такого типу будуть задачі синтезу логічних функцій, які еквівалентні даному дереву розпізнавання, оцінка їхньої складності, завдання мінімізації отриманого дерева. Досліджується складність граф-схемних моделей (структур логічних дерев класифікації), які конструюються у процесі навчання системи розпізнавання (логічне дерево класифікації фактично являє собою згенеровану функцію розпізнавання). Дана робота присвячена питанню оптимізації конструкції логічних дерев шляхом перестановки

структурних елементів в їх конструкції. Натепер існують підходи мінімізації (нормалізації) структур логічних дерев (ЛДК/АДК), які відрізняються алгоритмічною складністю та жорсткою спрямованістю під конкретні логічні граф-схемні представлення (класи логічних дерев). Метод перестановки ярусів (побідних блоків конструкції ЛДК) у структурі логічних дерев дозволяє добитися значного ефекту в оптимізації та може бути застосований для регулярного дерева довільної складності. Також варто зафіксувати суттєві переваги даного підходу мінімізації логічних дерев класифікації у плані програмної простоти побудови дерев класифікації, зменшення часу загальної генерації логічного дерева тощо. Робота актуальна для всіх методів розпізнавання образів, у яких отримана функція класифікації може бути представлена у вигляді логічного дерева.
Ключові слова: перестановка ярусів; логічне дерево; граф-схемні моделі; мінімізація логічних дерев.

ПОСТАНОВКА ЗАДАЧІ

Дана робота продовжує напрям досліджень із [2–5; 16–18] та присвячена загальному питанню оптимізації конструкції (структур) логічних дерев (моделей ЛДК/АДК) шляхом перестановки їхніх структурних елементів. Для всебічного дослідження концепції логічних дерев на першому етапі необхідно розглянути принципове питання представлення функції *k*-значної логіки у граф-схемній (графічній) формі. Як відомо, функції *k*-значної логіки, за аналогією із двозначною, можна представити в табличній, аналітичній та графічній формах. Головна увага буде приділена графічній формі представлення функцій *k*-значної логіки, ідея якої полягає в тому, що довільну *k*-значну логічну функцію можна представити у вигляді деякого зв'язного графа, який не містить циклів (у теорії графів такий вигляд графа називається деревом). Отже, назвемо вказане представлення логічної функції логічним деревом (або просто деревом).

Наприклад, на рис. 1 зображено логічне дерево, або граф функції тризначної логіки, яка залежить від трьох змінних, аргументів (зауважимо, що тут коло – x_1, трикутник – x_2, квадрат – x_3). Звернемо увагу на те, що вершини графа відповідають змінним, від яких залежить результуюча функція. У кожну вершину входить три ребра, які відповідають різним значенням змінних (0,12). Усього ребер нижнього ярусу графа – 3^n, тобто ця кількість дорівнює кількості різних наборів аргументів функції від n-змінних. Якщо рухатися від верхньої вершини графа вниз по ребрам до самого нижнього ярусу ребер графа, то кожний такий шлях буде відповідати визначеному набору змінних. Так, наприклад, нехай a, b, c на рис. 3.1 відповідає набору $\alpha = (1,0,2)$. Кожному кінцевому ребру такого шляху відповідає значення функцій на наборі, який відповідає фіксованому шляху, до якого належить дане ребро, наприклад, на наборі $\alpha = (1,0,2), f(x_1, x_2, x_3) = 2$.

У межах даного дослідження буде розглянуто принципове питання ефективної мінімізації регулярного логічного дерева за допомогою методу перестановки ярусів у його структурі.

АНАЛІЗ ОСТАННІХ ДОСЛІДЖЕНЬ І ПУБЛІКАЦІЙ

Дослідження продовжує цикл публікацій, які присвячені загальній проблематиці деревоподібних логічних структур (моделей логічних дерев) із погляду можливості їх використання в методах логічних дерев класифікації та розпізнавання. У роботах порушено загальні питання побудови, використання та мінімізації (оптимізації й обрізки) логічних дерев. Так, з робіт [6–8] відомо, що схема (правило) класифікації, яка побудована довіль-

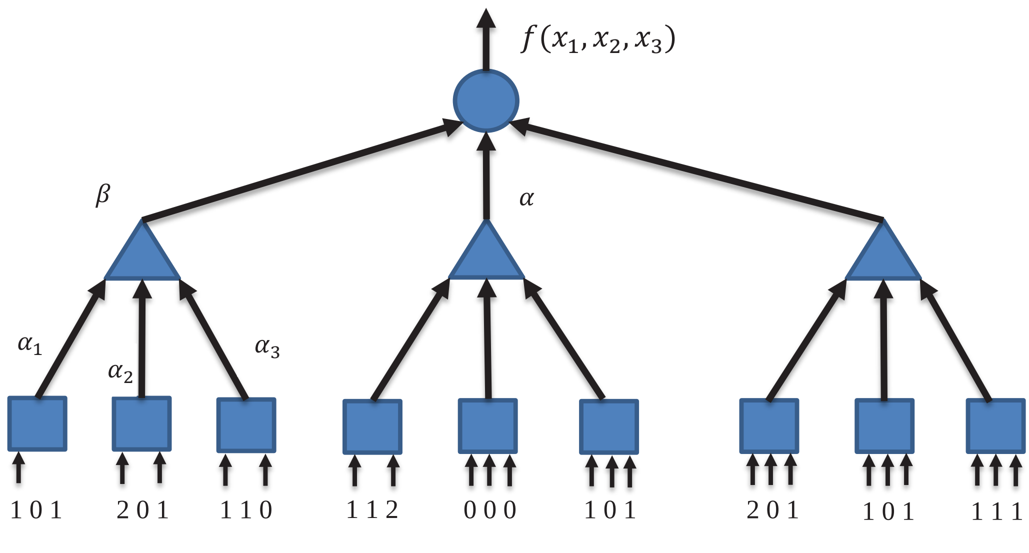

Рис. 1. Логічне дерево функції тризначної логіки $f(x_1, x_2, x_3)$ ярусів у його структурі

ним методом або алгоритмом розгалуженого вибору ознак (дерев класифікації), має деревоподібну логічну структуру. Причому така логічна структура (дерево) складається з вершин (атрибутів), які групуються за ярусами і які отримані на певному кроці (етапі) побудови дерева класифікації. Зазначимо, що важливою особливістю дерев класифікації є гнучкість, тобто здатність логічних дерев послідовно враховувати та досліджувати ефект впливу окремих змінних, атрибутів структури. Відповідно, є ще ціла низка причин, що забезпечують структурам логічних дерев більшу гнучкість, ніж традиційні методи й інструменти аналізу даних [9–12]. Так, здатність логічних дерев виконувати одномірне розгалуження для аналізу впливу (важливості, якості) окремих змінних дає можливість працювати зі змінними різних типів у вигляді предикатів (у випадку алгоритмічних дерев – відповідними автономними алгоритмами класифікації та розпізнавання) [13]. Принциповою задачею, яка виникає з роботи [14], є задача синтезу структур дерев класифікації, які будуть представлятися фактично деревом (графом) алгоритмів. На відміну від існуючих методів, головною особливістю деревоподібних систем розпізнавання є те, що важливість окремих ознак (групи ознак чи алгоритмів) визначається щодо функції, яка задає розбиття об'єктів на класи [15].

Відокремлення не вирішених раніше частин загальної проблеми

Зважаючи на загальний аналіз поточної проблематики моделей логічних дерев (структур логічних та алгоритмічних дерев класифікації), можна виділити проблему побудови ефективних методів та схем оптимізації (мінімізації та фінальної обрізки) логічних структур (структур ЛДК/АДК). Можливість ефективної й економної з погляду апаратних ресурсів перестановки ярусів конструкції логічних дерев класифікації (метод блокової мінімізації ЛДК) із метою отримання мінімальної структурної складності дерева класифікації.

МЕТА ДОСЛІДЖЕННЯ

Метою даного дослідження буде отримання ефективного способу (схеми) перестановки ярусів (структурних блоків) у конструкції фіксованого логічного дерева для мінімізації його структурної складності: кількості вершин, ярусів, листів, міток тощо.

Саме для досягнення даної мети поставлено такі завдання:

− дати оцінки рівню мінімізації конструкції логічного дерева (структури ЛДК/АДК);

− дослідити питання структурної складності логічного дерева для випадку регулярної структури;

− представити просту схему обрізки (мінімізації) структури логічного дерева шляхом перестановки блоків конструкції (ярусів).

ОСНОВНИЙ МАТЕРІАЛ

На першому етапі дослідження розглянемо принципове питання ефективної мінімізації регулярного логічного дерева за допомогою методу перестановки ярусів у його структурі. Так, з робіт [16–18] відомо, що логічне дерево також можна досить ефективно використовувати для мінімізації функції багатозначної логіки, яку воно представляє. Для демонстрації цього факту розглянемо такий приклад.

Приклад 1. Дана деяка логічна функція, яка записана в такому вигляді ($k = 3$):

$$f(x_1, x_2) = 1 * \varphi_0(x_1) \vee 1 * \varphi_1(x_2) \vee 1 * \varphi_2(x_2) \vee \varphi_2(x_1) * \varphi_2(x_2) .$$

За допомогою базових співвідношень:

$$P * \varphi_0(x) \vee P * \varphi_1(x) \vee \ldots \vee P * \varphi_{k-1}(x) = P;$$

$$P * 1 * \varphi_1(x) \vee P * 2 * \varphi_2(x) \vee \ldots \vee P * (k-1) * \varphi_{k-1}(x) = P * x$$

знаходимо відповідну ДДНФ:

$$1 * \varphi_0(x_1) * \varphi_0(x_2) \vee 1 * \varphi_0(x_1) * \varphi_1(x_2) \vee 1 * \varphi_0(x_1) * \varphi_2(x_2) \vee$$
$$\vee 1 * \varphi_1(x_1) * \varphi_1(x_2) \vee 1 * \varphi_2(x_1) * \varphi_1(x_2) \vee 1 * \varphi_1(x_1) * \varphi_2(x_2) \vee$$
$$\vee \varphi_2(x_1) * \varphi_2(x_2).$$

Якщо провести всі можливі склейки, то на виході отримаємо скорочену ДНФ вигляду – $1 * \varphi_0(x_1) \vee x_1 x_2 \vee 1 * x_2$. На наступному етапі складанням імплікативної таблиці або іншим шляхом знаходимо мінімальну та єдину (тупікову) ДНФ, яка має вигляд – $1 * \varphi_0(x_1) \vee x_1 x_2$.

Описаний метод, аналогічний відомому в булевій алгебрі методу Квайна, пов'язаний із досить громіздкими обчисленнями за великого числа змінних, фактично за $n > 3, k > 3$ не може мати практичного застосування.

Зауважимо, що інші методи (інтерпретації карт Карно та методи Мак-Класки для багатозначного випадку) також мають вказані вище недоліки.

Розглянемо застосування метода логічного дерева для даного прикладу. Так, загальна структура логічного дерева для заданої функції має такий вид (рис. 2).

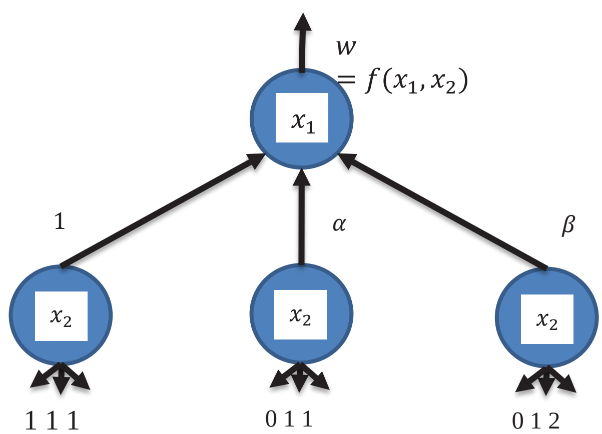

Рис. 2. Логічне дерево $f(x_1, x_2)$ із прикладу 1

Після заміни міток α, β та w через змінні x_1, x_2 маємо таке:

$$\alpha = 1 * \varphi_1(x_2) \vee 1 * \varphi_2(x_2) = 1 * (\varphi_1(x_2) \vee \varphi_2(x_2)) = 1 * x_2;$$

$$\beta = x_2;$$

$$w = f(x_1 x_2) = 1 * \varphi_0(x_1) \vee \alpha * \varphi_1(x_1) \vee \beta * \varphi_2(x_1) =$$

$$= 1 * \varphi_0(x_1) \vee \varphi_1(x_1) * 1 * x_2 \vee \varphi_2(x_1) * x_2 =$$

$$= 1 * \varphi_0(x_1) \vee x_2 * (1 * \varphi_1(x_1) \vee \varphi_2(x_1)) = 1 * \varphi_0(x_1) \vee x_1 x_2.$$

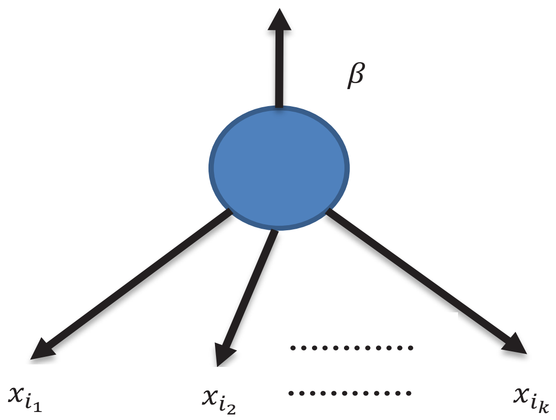

Рис. 3. Елементарне логічне дерево

Тобто отримали мінімальну форму початкової функції $f(x_1 x_2)$ значно більш швидким шляхом, ніж у разі використання попереднього методу.

Однак варто зауважити, що в даному прикладі не використовувався метод пошуку оптимального регулярного дерева, тобто дерево із природним розташуванням змінних x_1, x_2 вже дало мінімальну форму.

У разі використання концепції логічного дерева для цілей мінімізації важливим моментом другого етапу є розстановка функцій – міток (або просто міток) на структурі логічного дерева. Зазначимо, що будемо враховувати такі зауваження, які стосуються елементарного дерева, такого вигляду (рис. 3).

Тут $x_{i_j} \in \{0, 1, 2, \ldots, k-1\}$, $j = 1, 2, \ldots, k$ також $x_{i_1}, \ldots, x_{i_k}$ – деяке вхідна дія (вхідна функція), а β – вихідна функція (мітка):

1) якщо на вході $x_{i_1} = x_{i_2} = \ldots = x_{i_k} = \alpha$, то вихідна функція – мітка має вигляд $\alpha * \varphi_0(x) \vee \alpha * \varphi_1(x) \vee \ldots \vee \alpha * \varphi_{k-1}(x) = \alpha * (k-1) = \alpha$;

2) у разі входу $x_{i_1} = 0; x_{i_2} = 1; \ldots; x_{i_k} = k-1$ маємо:

$$\beta = 0 * \varphi_0(x) \vee 1 * \varphi_1(x) \vee \ldots \vee (k-1) * \varphi_{k-1}(x) = x;$$

3) у разі входу $x_{i_1} = k-1; x_{i_2} = k-2; \ldots; x_{i_k} = 0$ маємо таке:

$$\beta = (k-1) * \varphi_0(x) \vee (k-2) * \varphi_1(x) \vee .. \vee 0 * \varphi_{k-1}(x) = \bar{x} = k-1-x.$$

Зазначимо, що доведення цього факту випливає зі збігу правої та лівої частин за довільних значень x із множини $\{0, 1, 2, \ldots, k-1$;

4) нехай вхідна дія полягає в такому: $0, 1, \ldots, i-1, i, i, \ldots, i$. У такому разі будемо мати:

$$\beta = 0 * \varphi_0(x) \vee 1 * \varphi_1(x) \vee \ldots \vee i * \varphi_i(x) \vee i * \varphi_{i+1}(x) \vee \ldots \vee i * \varphi_{k-1}(x).$$

Помножимо перші $(i+1)$ членів на i. Очевидно, що водночас значення кон'юнкцій не зміниться, бо i – більше будь-якої із цих кон'юнкцій. Далі, починаючи з $(i+2)$ члена кожну кон'юнкцію множимо на індекс характеристичної функції, яка входить в дану кон'юнкцію. Після виконання вказаних перетворень будемо мати таке:

$$0 * i * \varphi_0(x) \vee 1 * i * \varphi_1(x) \vee \ldots \vee i * i * \varphi_i(x) \vee i * (i+1) * \varphi_{i+1}(x) \vee \ldots$$

$$\ldots \vee i * (k-1) * \varphi_{k-1}(x) = (0 * \varphi_0(x) \vee 1 * \varphi_1(x) \vee \ldots \vee i * \varphi_i(x) \vee$$

$$\vee (i+1) * \varphi_{i+1}(x) \vee \ldots \vee (k-1) * \varphi_{k-1}(x)) * i = i * x$$

(тут використовується пункт 2);

5) нехай вхідна дія полягає в такому: $k-1, k-2, \ldots, k-i-1, \ldots, k-i-1$. У такому разі будемо мати:

$$\beta = (k-1) * \varphi_0(x) \vee (k-2) * \varphi_1(x) \vee \ldots \vee (k-i) * \varphi_{i-1}(x) \vee$$

$$\vee (k-i-1) * \varphi_i(x) \vee \ldots \vee (k-i-1) * \varphi_{k-1}(x).$$

На наступному етапі представимо вихідну функцію за формулою (3.1):

$$(k-1) * \varphi_0(x) \vee (k-2) * \varphi_1(x) \vee \ldots \vee (k-i) * \varphi_{i-1}(x) \vee$$

$$\vee (k-i-1) * \varphi_i(x) \vee \ldots \vee (k-i-1) * \varphi_{k-1}(x),$$

значення $k-1, \ldots, k-i-1$ представимо у вигляді диз'юнкції:

$$k-1 \vee k-i-1 = k-1; k-i-1 \vee k-i-1 = k-i-1,$$

а значення $k-i-1$, починаючи з $(i+2)$-го члена, у вигляді:

$$k-i-1 \vee k-i-2 = k-i-1, \ldots, k-i-1 \vee 1 = k-i-1,$$

$$k-i-1 \vee 0 = k-i-1.$$

Далі підставимо ці значення у вираз вихідної функції, отримаємо такий вираз:

$$(k-1 \vee k-i-1) * \varphi_0(x) \vee (k-2 \vee k-i-1) * \varphi_1(x) \vee \ldots$$

$$\ldots \vee (k-i-1 \vee k-i-1) * \varphi_i(x) \vee (k-i-1 \vee k-i-2) * \varphi_{i+1}(x) \vee \ldots$$

$$\ldots \vee (k-i-1 \vee 1) * \varphi_{k-2}(x) \vee (k-i-1 \vee 0) * \varphi_{k-1}(x) =$$

$$= ((k-1 * \varphi_0(x) \vee (k-2) * \varphi_1(x) \vee \ldots \vee (k-i-1) * \varphi_i(x) \vee \ldots$$

$$\ldots \vee 1 * \varphi_{k-2}(x) \vee 0 * \varphi_{k-1}(x)) \vee (k-i-1) * (\varphi_0(x) \vee \varphi_1(x) \vee \ldots$$

$$\ldots \vee {*}\varphi_i(x) \vee \ldots \vee \varphi_{k-2}(x) \vee \varphi_{k-1}(x)) = \bar{x} \vee k-i-1;$$

6) нехай вхідна дія полягає в такому: $i, i, \ldots, i+1, i+2, \ldots, k-2, k-1$. У такому разі $\beta = x \vee i$.

Доведення. Вихідну функцію за формулою (3.1) представимо в такому вигляді:

$$i * \varphi_0(x) \vee i * \varphi_1(x) \vee \ldots \vee i * \varphi_i(x) \vee (i+1) * \varphi_{i+1}(x) \vee \ldots$$

$$\ldots \vee (k-1) * \varphi_{k-1}(x).$$

Значення i, яке входить в перші $i+1$ кон'юнкції, будуть представлені у вигляді диз'юнкцій зі значенням індексів, які відповідають характеристичним функціям. Значення членів не будуть мінятися, бо значення індексів менше, ніж i. Починаючи з $(i+2)$-го члена, кожний множник $i+1, i+2, \ldots, k-1$ представимо у вигляді диз'юнкції з i. На значення кон'юнкцій це перетворення також не впливає, бо i менше, ніж ці множники. Вихідну функцію в перетвореному вигляді представимо так: $x \vee i$;

7) нехай вхідними функціями є сигнали $i, i, \dots, i, k-1, k-2, \dots, i$ або $i, i, \dots, i, k-1, k-2, \dots, i+1$.

У цьому разі будемо мати таке:

$\beta = i \vee (k-1) * \varphi_{i+1}(x) \vee \dots \vee i * \varphi_{k-1}(x)$ або

$$\beta = i \vee (k-1) * \varphi_{i+1}(x) \vee \dots \vee (i+1) * \varphi_{k-1}(x).$$

Доведення. Вихідна функція конструюється так:

$$i * \varphi_0(x) \vee i * \varphi_1(x) \vee \dots \vee i * \varphi_i(x) \vee (k-1) * \varphi_{i+1}(x) \vee \dots \vee i * \varphi_{k-1}(x) =$$

$$i * \varphi_0(x) \vee i * \varphi_1(x) \vee \dots \vee i * \varphi_i(x) \vee ((k-1) \vee i) * \varphi_{i+1}(x) \vee \dots$$

$$\dots \vee (i \vee i) * \varphi_{k-1}(x) = i * (\varphi_0(x) \vee \varphi_1(x) \vee \dots \vee \varphi_i(x) \vee \varphi_{i+1}(x) \vee \dots$$

$$\dots \vee \varphi_{k-1}(x)) \vee (k-1) * \varphi_{i+1}(x) \vee \dots \vee i * \varphi_{k-1}(x) =$$

$$= i \vee (k-1) * \varphi_{i+1}(x) \vee \dots \vee i * \varphi_{k-1}(x).$$

Для вхідної функції $i, i, \dots, i, k-1, k-2, \dots, i+1$ доведення буде аналогічним;

нехай вхідними функціями є дія $k-1, k-2, \dots, i+1, i, i+1, \dots, k-2, k-1$. Вихідна функція представляється у вигляді $\beta = x \vee \bar{x}$.

Доведення. Вихідна функція конструюється так:

$$(k-1) * \varphi_0(x) \vee (k-2) * \varphi_1(x) \vee \dots \vee (i+1) * \varphi_{i-1}(x) \vee i * \varphi_0(x) \vee$$

$$\vee (i+1) * \varphi_{i+1}(x) \vee \dots \vee (k-1) * \varphi_{k-1}(x) = ((k-1) \vee 0) * \varphi_0(x) \vee$$

$$\vee ((k-2) \vee 1) * \varphi_1(x) \vee \dots \vee ((i+1) \vee (i-1)) * \varphi_{i-1}(x) \vee (i \vee i) * \varphi_i(x) \vee$$

$$\vee ((i+1) \vee (k-i-2)) * \varphi_{i+1}(x) \vee \dots \vee ((k-1) \vee 0) * \varphi_{k-1}(x).$$

Беремо до уваги дистрибутивний закон, маємо таке:

$$(0 * \varphi_0(x) \vee 1 * \varphi_1(x) \vee \dots \vee i * \varphi_i(x) \vee \dots \vee (k-1) * \varphi_{k-1}(x)) \vee$$

$$\vee ((k-1) * \varphi_0(x) \vee (k-2) * \varphi_1(x) \vee \dots \vee i * \varphi_i(x) \vee \dots \vee 0 * \varphi_{k-1}(x)) =$$

$$= x \vee \bar{x};$$

нехай вхідними функціями є дія $0, 1, \dots, i-1, i, i-1, \dots, 1, 0$.

Вихідна функція представляється у вигляді $x * \bar{x}$.

Доведення. Вихідна функція конструюється так:

$$0 * \varphi_0(x) \vee 1 * \varphi_1(x) \vee \dots \vee (i-1) * \varphi_{i-1}(x) \vee i * \varphi_i(x) \vee$$

$$\vee (i-1) * \varphi_{i+1}(x) \vee \dots \vee 0 * \varphi_{k-1}(x).$$

На наступному етапі всі члени цього виразу, до $(i+1)$-го включно, помножимо відповідно на значення $k-1, k-2, \dots, i$. Водночас значення даних кон'юнкцій не змінюється. Усі члени, починаючи з $(i+1)$-го місця, помножимо на індекс характеристичної функції, яка входить у дану кон'юнкцію. Після таких перетворень отримаємо такий вираз:

$$0 * (k-1) * \varphi_0(x) \vee 1 * (k-2) * \varphi_1(x) \vee \dots \vee (i-1) * (i+1) * \varphi_{i-1}(x) \vee$$

$$i * i * \varphi_i(x) \vee (i-1) * (i+1) * \varphi_{i+1}(x) \vee \dots \vee (k-2) * 1 * \varphi_{k-2}(x) \vee$$

$$\vee (k-1) * 0 * \varphi_{k-1}(x).$$

Зазначимо, що отриманий вираз можна представити так:

$$(0 * \varphi_0(x) \vee 1 * \varphi_1(x) \vee \dots \vee i * \varphi_i(x) \vee \dots \vee (k-1) * \varphi_{k-1}(x)) \wedge$$

$$\wedge ((k-1) * \varphi_0(x) \vee (k-2) * \varphi_1(x) \vee \dots \vee i * \varphi_i(x) \vee \dots$$

$$\dots \vee 0 * \varphi_{k-1}(x)) = x * \bar{x};$$

якщо вхідними функціями є така дія $i, \dots, i, i, i-1, \dots, 1, 0$, де i до i – місця, тоді вихідна функція буде представлена в такому вигляді: $\bar{x} \wedge i$.

Доведення. Вихідна функція за визначенням може бути представлена у вигляді:

$$i * \varphi_0(x) \vee i * \varphi_1(x) \vee \dots \vee i * \varphi_{i-1}(x) \vee (i-1) * \varphi_i(x) \vee (i-2) * \varphi_{i+1}(x) \vee \dots$$

$$\dots \vee 0 * \varphi_{k-1}(x) = i * (k-1) * \varphi_0(x) \vee i * (k-2) * \varphi_1(x) \vee \dots$$

$$\dots \vee (i+1) * i * \varphi_{i-1}(x) \vee (i-1) * i * \varphi_i(x) \vee (i-2) * i * \varphi_{i+1}(x) \vee \dots$$

$$\dots \vee 0 * i * \varphi_{k-1}(x) = ((k-1) * \varphi_0(x) \vee (k-2) * \varphi_1(x) \vee \dots$$

$$\dots \vee (i+1) * \varphi_{i-1}(x) \vee \dots \vee 0 * \varphi_{k-1}(x) \wedge i = \bar{x} \wedge i;$$

11 (a)) нехай вхідна дія буде в такому вигляді: $\underbrace{i, i, \dots, i}_{l}, \; i+j, i+j, \dots, i+j$ де $i+j \le k-1$.

У такому разі маємо:

$$\beta = i \vee (i+j) * \varphi_l(x) \vee (i+j) * \varphi_{l+1}(x) \vee \dots \vee (i+j) * \varphi_{k-1}(x),$$

за $i+j < k-1$, а якщо $i+j = k-1$, то в такому вигляді: $i \vee \varphi_l(x) \vee \dots \vee \varphi_{k-1}(x)$.

Доведення. Представимо дану вихідну функцію за допомогою початкової формули (3.1):

$$i * \varphi_0(x) \vee i * \varphi_1(x) \vee \dots \vee i * \varphi_{l-1}(x) \vee (i+j) * \varphi_l(x) \vee (i+j) * \varphi_{l+1}(x) \vee$$

$$\vee \dots \vee (i+j) * \varphi_{k-1}(x) = i * \varphi_0(x) \vee \dots \vee i * \varphi_{l-1}(x) \vee (i \vee (i+j)) * \varphi_l(x) \vee$$

$$\vee (i \vee (i+j)) * \varphi_{l+1}(x) \vee \dots \vee (i \vee (i+j)) * \varphi_{k-1}(x) = (i * \varphi_0(x) \vee \dots$$

$$\dots \vee i * \varphi_{l-1}(x) \vee i * \varphi_l(x) \vee i * \varphi_{l+1}(x) \vee \dots \vee i * \varphi_{k-1}(x)) \vee$$

$$\vee (i+j) * \varphi_l(x) \vee (i+j) * \varphi_{l+1}(x) \vee \dots \vee (i+j) * \varphi_{k-1}(x) =$$

$$= i * (\varphi_l(x) \vee \dots \vee \varphi_{l-1}(x) \vee \dots \vee \varphi_{k-1}(x)) \vee (i+j) * \varphi_l(x) \vee \dots$$

$$\dots \vee (i+j) * \varphi_{k-1}(x) = i \vee (i+j) * \varphi_l(x) \vee \dots \vee (i+j) * \varphi_{k-1}(x);$$

11 (b)) нехай вхідна дія буде така:

$$\underbrace{i, i, \dots, i}_{l}, \underbrace{i+j, i+j, \dots, i+j}_{m+1}, i, i, \dots, i.$$

Вихідна функція буде представлена в такому вигляді:

$\beta = i \vee (i+j) * \varphi_l(x) \vee \dots \vee (i+j) * \varphi_{l+m}(x)$;

11 (c)) нехай вхідна дія буде така:

$$\underbrace{i, i, \dots, i}_{l}, \underbrace{i+j, i+j, \dots, i+j}_{m+1}, i, i, \dots, i.$$

Вихідна функція буде представлена в такому вигляді:

$$\beta = (i+j) * \varphi_0(x) \vee \dots \vee (i+j) * \varphi_m(x) \vee i.$$

Зазначимо, що доведення зауважень (11 (b)) та (11 (c)) аналогічно доведенню (11 (a)).

Визначення 1. Нехай дана довільна функція двох змінних $f(x_1, x_2)$ багатозначної логіки. Трикутником будемо називати логічне дерево, яке можна побудувати для даної функції (рис. 4).

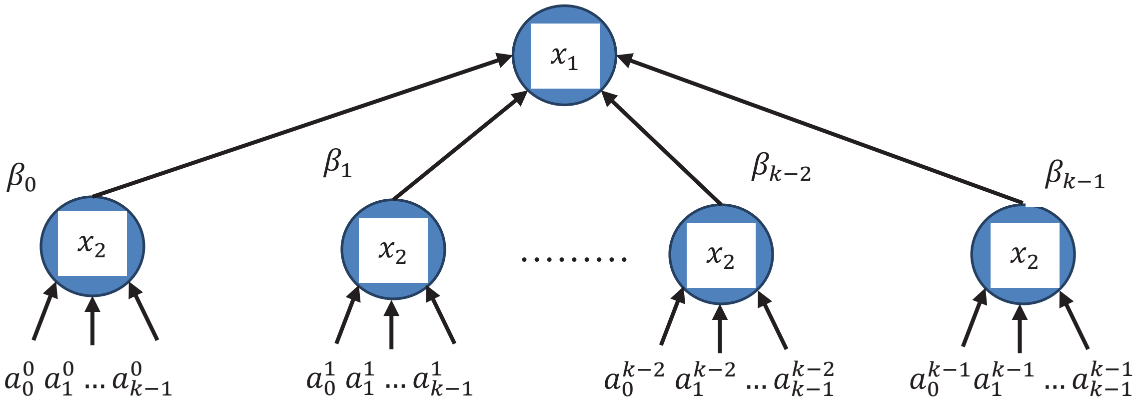

Рис. 4. Трикутник (логічне дерево) функції $\boldsymbol{f}(\boldsymbol{x}_1, \boldsymbol{x}_2)$

Теорема 1.

$$x_1 * (x_2 * (\alpha_0^0, \alpha_1^0, \dots, \alpha_{k-1}^0); x_2 * (\alpha_0^1, \alpha_1^1, \dots, \alpha_{k-1}^1); \dots$$

$$\dots; x_2 * (\alpha_0^{k-1}, \alpha_1^{k-1}, \dots, \alpha_{k-1}^{k-1})) = x_2 * (x_1 * (\alpha_0^0, \alpha_0^1, \dots, \alpha_0^{k-1});$$

$$; x_1 * (\alpha_1^0, \alpha_1^1, \dots, \alpha_1^{k-1}); \dots; x_1 * (\alpha_{k-1}^0, \alpha_{k-1}^1, \dots, \alpha_{k-1}^{k-1})).$$

Дана теорема визначає, що в разі перестановки ярусів логічного дерева, відповідних змінних x_1 та x_2, вхідні значення $\alpha_0^0, \dots, \alpha_{k-1}^0, \alpha_0^1, \alpha_1^1, \dots, \alpha_{k-1}^1, \alpha_{k-1}^1, \dots, \alpha_{k-1}^{k-1}$ міняються місцями та представляються у вигляді $\alpha_0^0, \dots, \alpha_0^{k-1}, \alpha_1^0, \dots, \alpha_{k-1}^0, \dots, \alpha_{k-1}^{k-1}$ (Рис. 4).

Доведення. Застосуємо теорему 1, функція $f(x_1, x_2)$ може бути представлена в такому вигляді:

$$f(x_1, x_2) = (\alpha_0^0 * \varphi_0(x_2) \vee \alpha_1^0 * \varphi_1(x_2) \vee \dots \vee \alpha_{k-1}^0 * \varphi_{k-1}(x_2)) * \varphi_0(x_1) \vee$$

$$\vee (\alpha_0^1 * \varphi_0(x_2) \vee \alpha_1^1 * \varphi_1(x_2) \vee \dots \vee \alpha_{k-1}^1 * \varphi_{k-1}(x_2)) * \varphi_1(x_1) \vee \dots$$

$$\dots \vee (\alpha_0^{k-1} * \varphi_0(x_2) \vee \alpha_1^{k-1} * \varphi_1(x_2) \vee \dots \vee \alpha_{k-1}^{k-1} * \varphi_{k-1}(x_2)) * \varphi_{k-1}(x_1).$$

На наступному етапі, після перетворення даного виразу за дистрибутивним законом, отримаємо таке:

$$(\alpha_0^0 * \varphi_0(x_1) \vee \alpha_0^1 * \varphi_1(x_1) \vee \dots \vee \alpha_0^{k-1} * \varphi_{k-1}(x_1)) * \varphi_0(x_2) \vee$$

$$\vee (\alpha_1^0 * \varphi_0(x_1) \vee \alpha_1^1 * \varphi_1(x_1) \vee \dots \vee \alpha_1^{k-1} * \varphi_{k-1}(x_1)) * \varphi_1(x_2) \vee \dots$$

$$\dots \vee (\alpha_{k-1}^0 * \varphi_0(x_1) \vee \alpha_{k-1}^1 * \varphi_1(x_1) \vee \dots \vee \alpha_{k-1}^{k-1} * \varphi_{k-1}(x_1)) * \varphi_{k-1}(x_2).$$

Зазначимо, що отриманий вираз представляє новий запис логічної функції $f(x_1, x_2)$, яка відповідає перестановці місцями ярусів змінних x_1 та x_2 в логічному дереві. Отриманий порядок вхідних функцій збігається з тим порядком, про який було сказано в теоремі 1. Саме цим фактом і доводиться справедливість даної теореми.

Припустимо, що задана багатозначна функція від n змінних. Побудуємо дерево даної функції. Розглянемо трикутник, який утворюється першим та другим каскадами (даний трикутник збігається з (Рис. 3) але вже $w = f(x_1, \dots, x_2)$) та трикутник, який отримується з нього перестановкою x_1 та x_2 (цей трикутник збігається з (Рис. 4)) але $w = f(x_1, x_2, \dots, x_n)$).

Оскільки мінімізація отримується зменшенням кількості різних функцій, то доцільно розглянути випадок, коли $\alpha_0^0 = \alpha_0^1 = \dots = \alpha_0^{k-1}$ або $\alpha_1^0 = \alpha_1^1 = \dots = \alpha_1^{k-1}, \dots, \alpha_{k-1}^0 = \dots = \alpha_{k-1}^{k-1}$, тому що за такої перестановки змінних можна отримати зменшення кількості різних функцій.

Для визначеності припустимо, що:

$$\alpha_0^0 = \alpha_1^0 = \dots = \alpha_0^{k-1}, \alpha_0^0 \neq \alpha_0^1 \neq \dots \neq \alpha_{k-1}^0, \dots, \alpha_0^{k-1} \neq \alpha_1^{k-1} \neq \dots \neq \alpha_{k-1}^{k-1}.$$

Зауважимо, що для всіх інших випадків загальна схема мінімізації буде аналогічною. У такому разі $\alpha_0^1 = \beta_0^1$. Зазначимо, що в дослідженні можуть виникнути декілька варіантів:

1) функції $\beta_0, \beta_1, \dots, \beta_{k-1}$ (Рис. 3) у нижніх ярусах дерева не траплялись, а отже, перестановкою змінних x_1 та x_2 замість k функцій $\beta_0, \beta_1, \dots, \beta_{k-1}$ отримаємо тільки $k-1$ функцій, серед яких можуть бути і такі, які вже траплялися на нижніх ярусах. Відповідно, перестановка x_1 та x_2 приводить до зменшення кількості різних функцій;

2) деякі m логічних функцій із $\beta_0, \beta_1, \dots, \beta_{k-1}; (1 \le m \le k-1)$ у нижніх ярусах дерева траплялися, тоді знову можливі два варіанти:

2 (a)) якщо в разі перестановки змінних x_1 та x_2 отримаємо кількість функцій $p, (p > m)$, які вже траплялись на нижніх ярусах, то тим самим досягнуть зменшення кількості різних функцій;

2 (b)) якщо в разі перестановки змінних x_1 та x_2 отримаємо, що функції $\beta_0, \beta_1, \dots, \beta_{k-1}$ (Рис. 4) у нижніх ярусах не траплялись, то перестановка не зменшує кількості різних функцій. Тоді змінні x_1 та x_2 переставляти не вигідно;

3) якщо всі функції $\beta_0, \beta_1, \dots, \beta_{k-1}$ структури логічного дерева (Рис. 3) у нижніх ярусах траплялись, тоді перестановка змінних x_1 та x_2 не може зменшити кількості різних функцій (а отже, і структурної складності логічного дерева, що мінімізується), тому вона буде невигідною щодо процедури мінімізації.

За допомогою сформульованої в даному дослідженні властивості трикутника можна так визначити алгоритм мінімізації логічного дерева (графа) довільної функції.

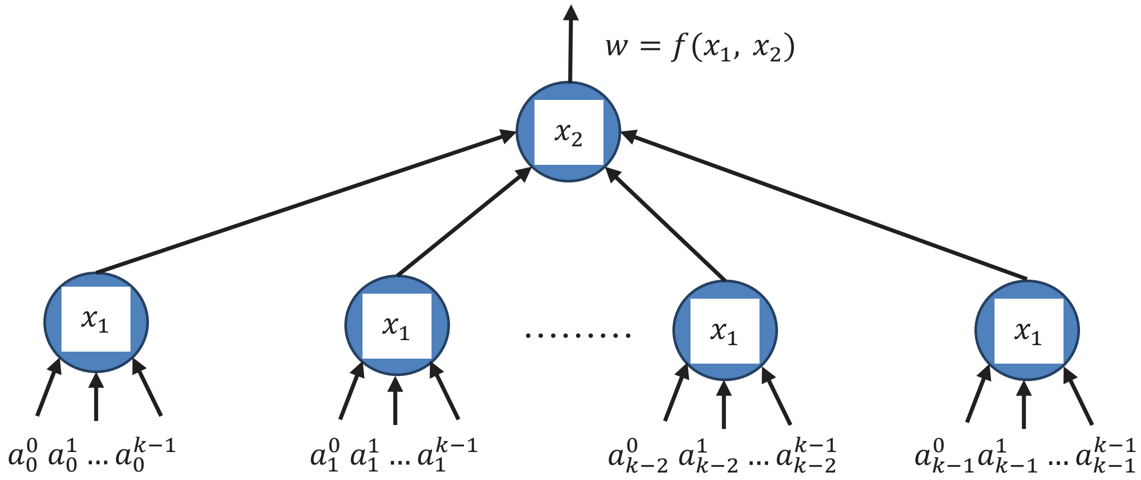

Рис. 5. Трикутник (логічне дерево) функції $\boldsymbol{f}(\boldsymbol{x}_1, \boldsymbol{x}_2)$ у разі перестановки ярусів, відповідних змінних $\boldsymbol{x}_1$ та $\boldsymbol{x}_2$

Нехай дана довільна логічна функція n змінних $f(x_1, x_2, \dots, x_n)$. Побудуємо для неї логічне дерево. Розглянемо трикутник, який утворюється першим та другим каскадами, з використанням властивостей трикутника, якщо потрібно, переставляємо змінні x_1 та x_2. На наступному етапі перевіряємо трикутники, які утворені другим та третім каскадами. Застосуємо аналогічні судження до кожного трикутника, спускаючись вниз за ярусами, доки не перевіримо всі трикутники. Отже, отримаємо такий алгоритм мінімізації за логічним деревом.

1) на першому етапі необхідно за табличною (або іншою) формою задання довільної логічної функції n змінних $f(x_1, x_2, \dots, x_n)$ побудувати відповідне логічне дерево;

2) далі на отриманій структурі логічного дерева треба провести розстановку відповідних міток (атрибутів) дерева та проаналізувати можливий ефект (у плані оптимізації структури дерева) від перестановки в його структурі ярусів (цим питанням буде приділена увага далі в межах даного дослідження);

3) на наступному етапі мінімізації, за допомогою теореми 3.1 (про розклад функції n змінних $f(x_1, x_2, \dots, x_n)$ за фіксованою змінною), необхідно представити логічну функцію через відповідні мітки (атрибути);

4) усі мітки логічного дерева відповідної функції n змінних $f(x_1, x_2, \dots, x_n)$ необхідно представити через змінні $x_1, x_2, \dots, x_n$, тобто отримаємо деяке дужкове представлення логічної функції.

В отриманому представленні логічної функції n змінних $f(x_1, x_2, \dots, x_n)$ за допомогою різних співвідношень проводяться процес подальшого спрощення (наприклад, 2-й дистрибутивний закон або інші) та дужкові перетворення.

З вищесказаного випливають такі переваги запропонованого простого метода мінімізації логічної функції щодо наявних методів:

1) простота роботи за вказаною схемою на відповідній графічній структурі (конструкції логічного дерева);

2) значна наочність запропонованого методу порівняно з іншими методами, є прямим або опосередкованим узагальненням двійкових (бінарних) методик для багатозначного випадку;

3) можливість нескладної програмної реалізації, що значно розширює сферу дій для великих n та k у разі багатозначної логіки;

4) природність запропонованого графічного представлення функцій багатозначної логіки (на основі концепції логічних дерев).

На даному етапі дослідження розглянемо такий приклад.

Приклад 2. Спростити функцію $f(x_1, x_2)$, задану таблично $(k = 5)$.

Побудуємо логічне дерево для даної функції (рис. 6).

$$\alpha = \varphi_3(x_2) \vee \varphi_4(x_2);$$

Таблиця 1. Таблична форма логічної функції $\boldsymbol{f}(\boldsymbol{x}_1, \boldsymbol{x}_2)$ прикладу 2

	f	0	0	0	4	4	1	1	0	3	4	2	2	0	2	2	3	3	0	1	1	4	4	0	0	0
	x_1	0	0	0	0	0	1	1	1	1	1	2	2	2	2	2	3	3	3	3	3	4	4	4	4	4
	x_2	0	1	2	3	4	0	1	2	3	4	0	1	2	3	4	0	1	2	3	4	0	1	2	3	4

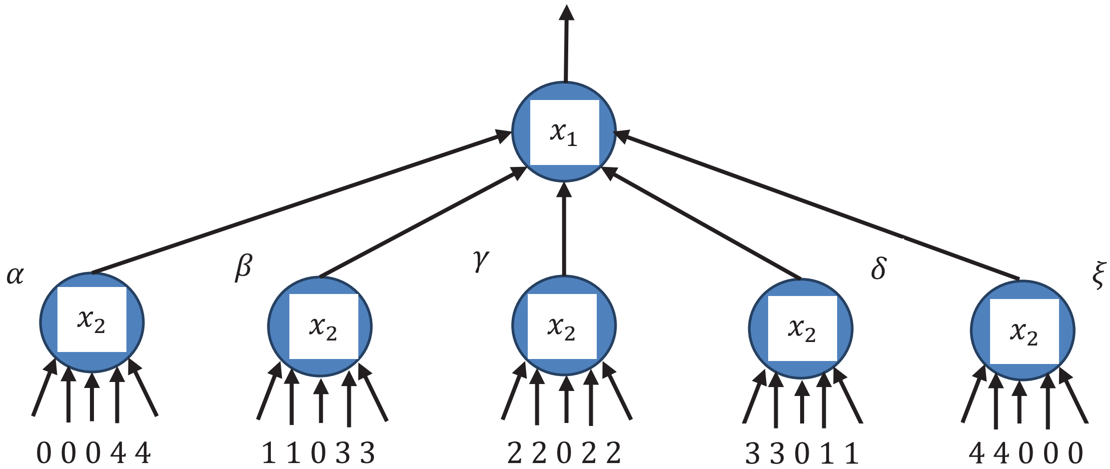

Рис. 6. Логічне дерево функції $f(x_1, x_2)$ прикладу 2

$$\varepsilon = \varphi_0(x_2) \vee \varphi_1(x_2);$$

$$\beta = 1 * \varphi_0(x_2) \vee 1 * \varphi_1(x_2) \vee 3 * \varphi_3(x_2) \vee 3 * \varphi_4(x_2);$$

$$\gamma = 2 * \varphi_0(x_2) \vee 2 * \varphi_1(x_2) \vee 2 * \varphi_3(x_2) \vee 2 * \varphi_4(x_2);$$

$$\delta = 3 * \varphi_0(x_2) \vee 3 * \varphi_1(x_2) \vee 1 * \varphi_3(x_2) \vee 1 * \varphi_4(x_2);$$

$$w = \alpha * \varphi_0(x_1) \vee \beta * \varphi_1(x_1) \vee \gamma * \varphi_2(x_1) \vee \delta * \varphi_3(x_1) \vee \xi * \varphi_4(x_1);$$

$$w = \varphi_3(x_2) * \varphi_0(x_1) \vee \varphi_4(x_2) * \varphi_0(x_1) \vee 1 * \varphi_0(x_2) * \varphi_1(x_1) \vee$$
$$\vee 1 * \varphi_1(x_2) * \varphi_1(x_1) \vee 3 * \varphi_3(x_2) * \varphi_1(x_1) \vee 3 * \varphi_4(x_2) * \varphi_1(x_1) \vee$$
$$\vee 2 * \varphi_0(x_2) * \varphi_2(x_1) \vee 2 * \varphi_1(x_2) * \varphi_2(x_1) \vee 2 * \varphi_4(x_2) * \varphi_2(x_1) \vee$$
$$\vee 3 * \varphi_0(x_2) * \varphi_3(x_1) \vee 3 * \varphi_1(x_3) * \varphi_3(x_1) \vee 1 * \varphi_3(x_2) * \varphi_3(x_1) \vee$$
$$\vee 1 * \varphi_4(x_2) * \varphi_2(x_1) \vee \varphi_0(x_2) * \varphi_4(x_1) \vee \varphi_1(x_2) * \varphi_4(x_1).$$

Зазначимо, що дане представлення початкової логічної функції двох змінних $f(x_1, x_2)$ не є її мінімальною ДНФ, бо можна провести ще додатково операцію поглинання.

Тому на наступному етапі побудуємо та розглянемо таку структуру логічного дерева після проведення операції перестановки змінних x_1 та x_2 (рис. 7):

$$\alpha = x_1; \beta = \overline{x_1}; w = x_1 * \varphi_0(x_2) \vee x_1 * \varphi_1(x_2) \vee \overline{x_1} * \varphi_3(x_2) \vee \overline{x_1} * \varphi_4(x_2).$$

Далі проведемо процедури мінімізації даної логічної функції, для цього представимо логічну функцію у ДДНФ та проведемо відповідні перетворення:

$$f(x_1, x_2) = 4 * \varphi_0(x_1) * \varphi_3(x_2) \vee 4 * \varphi_0(x_1) * \varphi_4(x_2) \vee 1 * \varphi_1(x_1) * \varphi_0(x_2) \vee$$
$$\vee 1 * \varphi_1(x_1) * \varphi_1(x_2) \vee 3 * \varphi_1(x_1) * \varphi_3(x_2) \vee 3 * \varphi_1(x_1) * \varphi_2(x_2) \vee$$
$$\vee 2 * \varphi_2(x_1) * \varphi_0(x_2) \vee 2 * \varphi_2(x_1) * \varphi_1(x_2) \vee 2 * \varphi_2(x_1) * \varphi_3(x_2) \vee$$
$$\vee 2 * \varphi_2(x_1) * \varphi_4(x_2) \vee 3 * \varphi_3(x_1) * \varphi_0(x_2) \vee 3 * \varphi_3(x_1) * \varphi_1(x_2) \vee$$
$$\vee 1 * \varphi_3(x_1) * \varphi_3(x_2) \vee 1 * \varphi_3(x_1) * \varphi_4(x_2) \vee 4 * \varphi_4(x_1) * \varphi_0(x_2) \vee$$
$$\vee 4 * \varphi_4(x_1) * \varphi_1(x_2) = (1 * \varphi_1(x_1) \vee 2 * \varphi_2(x_1) \vee 3 * \varphi_3(x_1) \vee$$

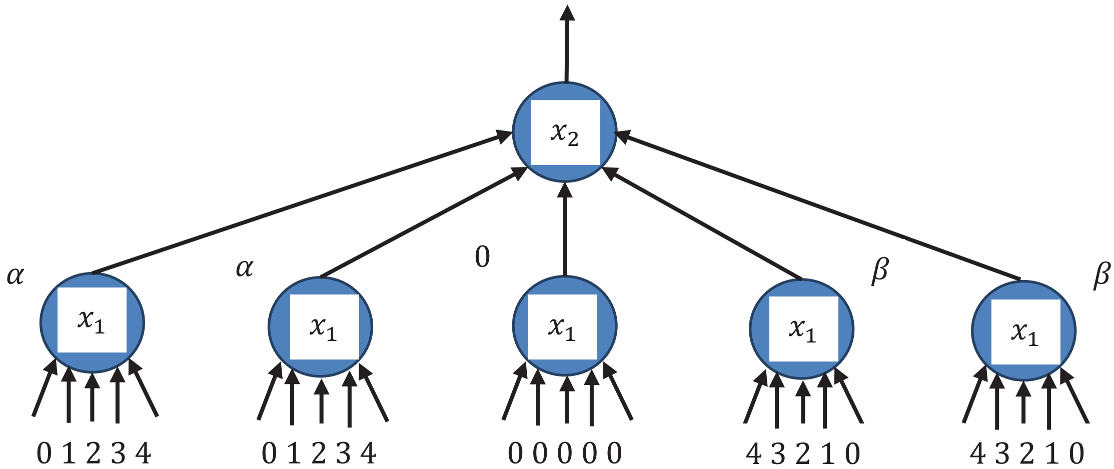

Рис. 7. Логічне дерево функції $f(x_1, x_2)$ прикладу 2 після перестановки змінних x_1 та x_2

$$\vee 4 * \varphi_4 (x_1)) * \varphi_0 (x_2) \vee (1 * \varphi_1 (x_1) \vee 2 * \varphi_2 (x_1) \vee 3 * \varphi_3 (x_1) \vee$$

$$\vee 4 * \varphi_4 (x_1)) * \varphi_1 (x_2) \vee (4 * \varphi_0 (x_1) \vee 3 * \varphi_1 (x_1) \vee 2 * \varphi_2 (x_1) \vee 1 * \varphi_3 (x_1)) *$$

$$* \varphi_3 (x_2) \vee (4 * \varphi_0 (x_1) \vee 3 * \varphi_1 (x_1) \vee 2 * \varphi_2 (x_1) \vee 1 * \varphi_3 (x_1)) * \varphi_4 (x_2) =$$

$$= x_1 * \varphi_0 (x_2) \vee x_1 * \varphi_1 (x_2) \vee \overline{x_1} * \varphi_3 (x_2) \vee \overline{x_1} * \varphi_4 (x_2).$$

Отже, на виході отримаємо такий вираз:

$$f (x_1, x_2) = x_1 * \varphi_0 (x_2) \vee x_1 * \varphi_1 (x_2) \vee \overline{x_1} * \varphi_3 (x_2) \vee \overline{x_1} * \varphi_4 (x_2).$$

ВИСНОВКИ

Зважаючи на все вищесказане, зафіксуємо такі пункти:

1) актуальним залишається питання ефективної мінімізації регулярного логічного дерева (моделі ЛДК) за допомогою методу перестановки ярусів у його структурі, бо логічне дерево також можна досить ефективно використовувати для мінімізації функції багатозначної логіки, яку воно представляє;

2) запропонований метод мінімізації на основі концепції логічних дерев можна досить ефективно використовувати для мінімізації функції багатозначної логіки, а отже, забезпечувати мінімальну форму дерева класифікації, яке вона представляє. Підкреслимо також, що даний метод дозволяє просту й ефективну програмну реалізацію, що значно розширює сферу дій для великих n та k у разі багатозначної логіки;

3) важливим моментом утакому методі мінімізації є пошук серед $n!$ логічних дерев (логічних структур) найкращого (оптимального в деякому сенсі) дерева, тобто логічного дерева, яке дає мінімальну або близьку до неї форму;

4) зауважимо, що загальна схема пошуку оптимального логічного дерева, яке дає мінімальну або близьку до неї форму (логічну структуру), значно простіше, ніж застосовування інших актуальних методів мінімізації функцій багатозначної логіки.

REFERENCES

[1] Zheng Z., Kohavi R., Mason L. (2001). *Real world performance of association rule algorithms* [Associative algorithms in logical trees]. *Proceedings of the Seventh ACM SIGKDD International Conference on Knowledge Discovery and Data Mining.* Ed. by F. Provost, R. Srikant. P. 401–406.

[2] Vasilenko Y. A., Vasilenko E. Y., Povhan I. F., Vashchuk F. G. (2004). *Conceptual basis of pattern recognition systems based on the method of branched feature selection*. [General concept of logical trees]. *Scientific and technical journal "European Journal of Enterprise Technologies".* 7 [1]. P. 13–15.

[3] Vasilenko Y. A., Vasilenko E. Y., Povhan I. F., Vashchuk F. G. (2003). *Method of branched feature selection in mathematical design of multilevel pattern recognition systems.* [Methods and algorithms for constructing logical trees]. *Scientific and technical journal "Artificial Intelligence".* № 7. P. 246–249.

[4] Vasilenko Y. A., Povhan I. F., Vashchuk F. G. (2011). *The problem of evaluation of complexity of logic trees, recognition, and a general method of optimization*. [Methods of optimization of logical trees]. *Scientific and technical journal "European Journal of Enterprise Technologies".* № 6/4 (54). P. 24–28.

[5] Povkhan I. F., Vashchuk F. G. (2012). *Overall assessment of minimization of logical tree structures* [The problem of minimizing logical trees]. *Scientific and technical journal "European Journal of Enterprise Technologies".* № 1/4 (55). P. 29–33.

[6] Quinlan J. R. (2008). *Induction of Decision Trees.* [General theory of logical trees]. *Machine Learning.* № 1. P. 1–81. 22.

[7] Vtogoff P. E. (2009). *Incremental Induction of Decision Trees.* [Algorithms of decision trees]. *Machine Learning.* № 4. P. 161–186.

[8] Laver V. O., Povkhan I. F. (2019). The algorithms for constructing a logical tree of classification in pattern recognition problems. [General concept of logical trees]. *Scientific notes of the Tauride national University. Series: technical Sciences.* Vol. 30 (69) № 4. P. 100–106.

[9] Srikant R., Agrawal R. (1997). Mining generalized association rules. [Association rules]. *Future Generation Computer Systems.* Vol. 13, № 2. P. 161–180.

[10] Hastie T., Tibshirani R., Friedman J. (2008). *The Elements of Statistical Learning*. P. 768.

[11] Povkhan I. F. (2018). *The problem of functional evaluation of the training sample in the problems of recognition of discrete objects.* [The problem of assessing the importance of features]. *Scientific notes of Taurida national University. Series: technical Sciences.* Volume 29 (68). № 6. P. 217–222.

[12] Povkhan I. F. (2019). Features of synthesis of generalized features in the construction of recognition systems using the logical tree method. *Materials of the international scientific and practical conference "Information technologies and computer modeling ITKM-2019".* P. 169–174.

[13] Povhan I. Designing of recognition system of discrete objects. (2016). [Classification tree methods]. *2016 IEEE First International Conference on Data Stream Mining & Processing (DSMP)*. Lviv, Ukraine. P. 226–231.

[14] Vasilenko Y. A., Vasilenko E. Y., Povkhan I. F. (2002). [A feature in pattern recognition theory]. Defining the concept of a feature in pattern recognition theory. *Scientific and technical journal "Artificial Intelligence".* № 4, P. 512–517.

[15] Subbotin S. A. (2019). [Decision trees]. Construction of decision trees for the case of low-information features. *Radio Electronics, Computer Science, Control*, № 1, P. 121–130.

[16] Povhan I. (2019). [Logical classification trees]. General scheme for constructing the most complex logical tree of classification in pattern recognition discrete objects. *Collection of scientific papers "Electronics and information technologies".* Lviv, Vol. 11. P. 73–80.

[17] Povhan I. (2019). [Logical classification trees]. Generation of elementary signs in the general scheme of the recognition system based on the logical tree. *Collection of scientific papers "Electronics and information technologies".* Lviv, Vol. 12. P. 20–29.

[18] Povhan I. (2020). [Logical classification trees]. Question of the optimality criterion of a regular logical tree based on the concept of similarity. *Collection of scientific papers "Electronics and information technologies".* Lviv, Vol. 13. P. 19–27.

СПИСОК ВИКОРИСТАНОЇ ЛІТЕРАТУРИ

[1] Zheng Z., Kohavi R., Mason L. Real world performance of association rule algorithms. *Proceedings of the Seventh ACM SIGKDD International Conference on Knowledge Discovery and Data Mining* / Ed. by F. Provost, R. Srikant. 2001. P. 401–406.

[2] Концептуальна основа систем розпізнавання образів на основі метода розгалуженого вибору ознак / Ю. А. Василенко та ін. *European Journal of Enterprise Technologies :* науково-технічний журнал. 2004. № 7 (1). **С**. 13–15.

[3] Метод розгалуженого вибору ознак в математичному конструюванні багаторівневих систем розпізнавання образів / Ю. А. Василенко та ін. *Штучний Інтелект :* науково-технічний журнал. 2003. № 7. С. 246–249.

[4] Проблема оцінки складності логічних дерев розпізнавання та загальний метод їх оптимізації / Ю. А. Василенко та ін. *European Journal of Enterprise Technologies :* науково-технічний журнал. 2011. № 6/4 (54). С. 24–28.

[5] Повхан І. Ф., Ващук Ф. Г. Загальна оцінка мінімізації деревоподібних логічних структур. *European Journal of Enterprise Technologies :* науково-технічний журнал. 2012. № 1/4 (55). С. 29–33.

[6] Quinlan J. R. Induction of Decision Trees. *Machine Learning.* 2008. № 1. P. 1–81.

[7] Vtogoff P. E. Incremental Induction of Decision Trees. *Machine Learning.* 2009. № 4. P. 161–186.

[8] Лавер В. О., Повхан І. Ф. Алгоритми побудови логічних дерев класифікації в задачах розпізнавання образів. *Вчені записки Таврійського національного університету.* Серія «Технічні науки». 2019. Т. 30 (69). № 4. С. 100–106.

[9] Srikant R., Agrawal R. Mining generalized association rules. *Future Generation Computer Systems.* 1997. Vol. 13. № 2. P. 161–180.

[10] Hastie T., Tibshirani R., Friedman J. The Elements of Statistical Learning. 2008. P. 768.

[11] Повхан І. Ф. Проблема функціональної оцінки навчальної вибірки в задачах розпізнавання дискретних об'єктів. *Вчені записки Таврійського національного університету.* Серія «Технічні науки». 2018. Т. 29 (68). № 6. С. 217–222.

[12] Повхан І. Ф. Особливості синтезу узагальнених ознак при побудові систем розпізнавання за методом логічного дерева. *Інформаційні технології та комп'ютерне моделювання ІТКМ-2019 :* матеріали Міжнародної науково-практичної конференції. Івано-Франківськ. 2019. С. 169–174.

[13] Povhan I. Designing of recognition system of discrete objects. *2016 IEEE First International Conference on Data Stream Mining & Processing (DSMP)*. Lviv, Ukraine, 2016. P. 226–231.

[14] Василенко Ю. А., Василенко Е. Ю., Повхан І. Ф. Визначення поняття ознаки в теорії розпізнавання образів. *Штучний Інтелект :* науково-технічний журнал. 2002. № 4. С. 512–517.

[15] Суботин С. А. Построение деревьев решений для случая малоинформативных признаков. *Radio Electronics, Computer Science, Control.* 2019. № 1. С. 121–130.

[16] Povhan I. F. General scheme for constructing the most complex logical tree of classification in pattern recognition discrete objects. *Електроніка та інформаційні технології :* збірник наукових праць. Lviv, 2019. Vol. 11. P. 73–80.

[17] Povhan I. F. Generation of elementary signs in the general scheme of the recognition system based on the logical tree. *Електроніка та інформаційні технології :* збірник наукових праць. Lviv, 2019. Vol. 12. P. 20–29.

[18] Povhan I. F. Question of the optimality criterion of a regular logical tree based on the concept of similarity. *Електроніка та інформаційні технології :* збірник наукових праць. Lviv, 2020. Vol. 13. P. 19–27.

Дата надходження статті до редакції: 09.10.2020
Дата затвердження статті до друку: 22.10.2020