УДК 519.718
DOI: 10.15587/1729-4061.2021.225325

Розробка алгоритму мінімізації булевих функцій для візуально-матричної форми аналітичного методу

М. Т. Соломко

Проведеними дослідженнями встановлена можливість збільшення ефективності візуально-матричної форми аналітичного методу мінімізації булевих функцій шляхом виявлення резервів більш складнішого алгоритму проведення логічних операцій поглинання та супер-склеювання змінних у термах логічних функцій.

Встановлено також збільшення ефективності процедури мінімізації булевих функцій шляхом вибору, за встановленими критеріями, оптимального стеку логічних операцій для першої та другої бінарних матриць булевих функцій. При комбінуванні послідовності логічних операцій з використанням різних способів склеювання змінних – простого та супер-склеювання існує невелике число випадків, коли мінімізація функції є більш ефективна, якщо у першій матриці спочатку застосувати операцію простого склеювання змінних. Таким чином, необхідний короткий аналіз для першочергового застосування операцій у першій бінарній матриці. Це забезпечує належну ефективність мінімізації до раніш не врахованих варіантів спрощення булевих функцій візуально-матричною формою аналітичного методу. Для ряду випадків вибір оптимального стеку потрібний і для другої бінарної матриці.

Експериментальними дослідженнями підтверджено, що візуально-матрична форма аналітичного методу, особливістю якої є використання систем 2-(n, b)-design та 2-(n, x/b)-design у першій матриці, підвищує ефективність процесу та достовірність результату мінімізації булевих функцій. При цьому спрощується процедура пошуку мінімальної функції. У порівнянні з аналогами це дає змогу підвищити продуктивність процесу мінімізації булевих функцій на 100–200 %.

Є підстави стверджувати про можливість збільшення ефективності процесу мінімізації булевих функцій візуально-матричною формою аналітичного методу, шляхом використання більш складних логічних операцій поглинання та супер-склеювання змінних. А також за допомогою оптимального комбінування послідовності логічних операцій супер-склеювання змінних та простого склеювання змінних, на підставі вибору, за встановленими критеріями, стеку логічних операцій у першій бінарній матриці заданої функції.

Ключові слова: мінімізація булевих функцій, візуально-матрична форма аналітичного методу, бінарна матриця.

1. Вступ

Аналітичний метод, в основу якого покладено рівносильні перетворення за допомогою законів і тотожностей булевої алгебри, ефективний при спрощенні порівняно простих булевих функцій. Розмова йде про те, щоб перейти від ДДНФ (ДКНФ) до ДНФ (КНФ) з мінімальним числом термів. Кількість літералів у кожному термі теж має бути мінімальна. Недоліком аналітичного методу є невизначеність послідовності логічних операцій при спрощенні функцій, і, отже, відсутність алгоритму мінімізації булевих функцій. У свою чергу відсутність алгоритму не завжди гарантує, що отриманий вираз булевої функції буде мінімальним і подальше його спрощення неможливе.

Дві форми інформації, які визначаються рефлективним та континуальним мисленням, представлено у роботі [1].

У першому випадку людина отримує інформацію словами, думає словами, та інколи перетворює їх в образи. Такий спосіб передачі інформації (вербальний) має малу інформаційну ємність, вимагає активної участі мозкових структур для розшифровки, переробки та доповнення прийнятої інформації.

При континуальній свідомості мислення відбувається не словами, а образами. Образне мислення характеризується великим надходженням інформації за одиницю часу, неспівставним з вербальним.

У роботі [2] зазначається, що «інформацію несуть не тільки усипані буквами листи книги або людська мова, але і сонячне світло, складки гірського хребта, шум водоспаду, шелест трави».

Візуально-матричні методи мінімізації булевих функцій розглядалися у працях [3–6] та ін.

Вербальна форма інформації – найбільш поширений варіант представлення опису предметної області. Будь-який алгебричний вираз – це насамперед текст, утворений за визначеними правилами. Алгебричний спосіб мінімізації булевих функцій є вербальною процедурою. До ілюстрацій звертаються у випадку коли є намагання пояснити те, що важко викласти текстом.

Ілюстративний (образний) опис наглядний, що дозволяє подати одночасно систему відношень між окремими змінними задачі. Характерною особливістю образу є його смислова ємність, здатність передавати великий об'єм інформації невеликою кількістю знаків і, як наслідок, забезпечення імпліціювання значної частини цієї інформації [7].

Таким чином, образна форма інформації у вигляді комбінаторних об'єктів має забезпечити більше шансів визначити алгоритм аналітичного методу мінімізації булевих функцій. Комбінаторними об'єктами у цьому випадку будуть двовимірні бінарні матриці та присутність у структурі таблиці істинності повних 2-(n, b)-design, або неповних 2-(n, x/b)-design систем з повторенням і суть власне комбінаторних образів [8–10]. У підсумку вербальні процедури алгебричних перетворень замінюються рівносильними образними перетвореннями [8].

Приклад 1. Методом Блейка-Порецького мінімізувати функцію $f(x_1, x_2, x_3, x_4)$ (1) [11].

$$f\left(x_1, x_2, x_3, x_4\right) = \overline{x}_1 x_2 + x_2 \overline{x}_3 x_4 + \overline{x}_1 x_3 x_4 + x_2 x_3 x_4. \qquad (1)$$

Метод Блейка-Порецького ґрунтується на застосуванні логічної операції узагальнене склеювання змінних

$$A \cdot x + B \cdot \bar{x} = A \cdot x + B \cdot \bar{x} + A \cdot B,$$

що дозволяє знаходити мінімальну функцію за довільною ДНФ її представлення.

Друга та четверта змінні функції $f(x_1, x_2, x_3, x_4)$ (1) допускають узагальнене склеювання за змінною x_3.

$$x_2 \overline{x_3} x_4 + x_2 x_3 x_4 = x_2 \overline{x_3} x_4 + x_2 x_3 x_4 + x_2 x_4.$$

Очевидно, що ніякі інші змінні функції $f(x_1, x_2, x_3, x_4)$ (1) не допускають узагальнене склеювання за іншими змінними.

Виконавши останнє узагальнене склеювання, отримаємо:

$$f(x_1, x_2, x_3, x_4) = \overline{x_1} x_2 + x_2 \overline{x_3} x_4 + \overline{x_1} x_3 x_4 + x_2 x_3 x_4 + x_2 x_4.$$

Після виконання всіх поглинань отримаємо мінімальну функцію:

$$f(x_1, x_2, x_3, x_4) = \overline{x_1} x_2 + \overline{x_1} x_3 x_4 + x_2 x_4. \qquad (2)$$

Подальше застосування операцій узагальнене склеювання змінних та поглинань не дають результату.

Мінімізація функції $f(x_1, x_2, x_3, x_4)$ (1) [11] рівносильними образними перетвореннями має такий вигляд:

$$f(x_1, x_2, x_3, x_4) = \begin{array}{|cccc|} \hline x_1 & x_2 & x_3 & x_4 \\ \hline 0 & 1 & & \\ & 1 & 0 & 1 \\ 0 & & 1 & 1 \\ & 1 & 1 & 1 \\ \hline \end{array} =$$

$$= \begin{vmatrix} 0 & 1 & & \\ 0 & & 1 & 1 \\ & 1 & & 1 \end{vmatrix} = \overline{x_1} x_2 + \overline{x_1} x_3 x_4 + x_2 x_4.$$

Мінімальна функція отримана за допомогою операції простого склеювання змінних:

$$f(x_1, x_2, x_3, x_4) = \overline{x_1} x_2 + \overline{x_1} x_3 x_4 + x_2 x_4. \qquad (3)$$

Спростити функцію (1) можна і за допомогою операції узагальнене склеювання змінних:

$$
f\left(x_1,x_2,x_3,x_4\right)=\begin{array}{|cccc|}
\hline
x_1 & x_2 & x_3 & x_4 \\
\hline
0 & 1 & & \\
 & \color{red}{1} & \color{red}{0} & \color{red}{1} \\
0 & & 1 & 1 \\
 & \color{red}{1} & \color{red}{1} & \color{red}{1} \\
\hline
\end{array}
=\left|\begin{array}{cccc}
0 & 1 & & \\
 & \color{red}{1} & \color{red}{0} & \color{red}{1} \\
0 & & 1 & 1 \\
 & \color{red}{1} & \color{red}{1} & \color{red}{1} \\
 & \color{red}{1} & & \color{red}{1}
\end{array}\right|=
$$

$$
=\left|\begin{array}{cccc}
0 & 1 & & \\
0 & & 1 & 1 \\
 & 1 & & 1
\end{array}\right|=\overline{x}_1x_2+\overline{x}_1x_3x_4+x_2x_4.
$$

Чергування нулів і одиниць у третьому стовпчику першої матриці є гіперпарамент (більше ніж передумова) для операції узагальнене склеювання, за змінною x_3. У другій матриці проведена операція поглинання змінних. Результати мінімізації (2) і (3) співпадають, однак метод образних перетворень є суттєво простішою процедурою спрощення функції.

Візуально-матричний спосіб мінімізації булевих функцій – це до певної міри, завершений і самостійний метод, який ґрунтується на використанні деяких властивостей візуального сприйняття інформації [3–6]. Найбільш компактною формою представлення інформації у цьому випадку є двовимірна матриця. Порядок взаємного розташування елементів матриці, однаковий при алгебричному підході, відіграє суттєву роль при візуальному сприйнятті двовимірних даних. Потенційні можливості мінімізації булевих функцій, представлених за допомогою матриці, забезпечуються властивостями періодичності та симетрії зображення [3, 5]. Хоча візуально-матричні методи стали відомі з кінця 1940 – початку 1950-х рр., візуальний метод, який ґрунтується на бінарних комбінаторних системах з повторенням 2-(n, b)-design, 2-(n, x/b)-design, як дещо окреме від періодичності та симетрії матричного зображення, продукт конкретного припущення, започаткований і розвивається з 2017 р. [8].

У роботі [12] розглянуто послідовне чергування логічних операцій суперсклеювання змінних (якщо така операція можлива) та простого склеювання змінних у першій матриці булевої функції, що забезпечує ефективність мінімізації та є основою алгоритмізації аналітичного методу. Продемонстровано також приклад мінімізації 4-розрядної булевої функції, коли послідовне застосування зазначених логічних операцій не завжди є оптимальним за ефективністю процедури.

Еволюція аналітичного методу спрощення логічних функцій та його алгоритмізація є результатом невпинної оптимізації, тому актуальними залишаються дослідження направлені, зокрема, на внесення оновлення до алгоритму мінімізації аналітичним методом, який розглянутий у роботі [12]. Це дасть змогу забезпечити належну ефективність до раніш не врахованих варіантів спрощення булевих функцій аналітичним методом, зокрема у класі ДДНФ та ДКНФ представленні, а також оптимізувати вартість технології мінімізації булевих функцій аналітичним методом.

2. Аналіз літературних даних та постановка проблеми

Новий метод мінімізації логічних функцій з порівняно малим числом змінних запропоновано у роботі [13]. Метод передбачає включення упорядкування оптимальних мінтермів логічної функції. Обрані мінтерми та процедура мінімізації забезпечують оптимальне покриття. Цей вид оптимального упорядкування допомагає алгоритму мінімізації, швидко охоплювати потрібні мінтерми, що дає змогу зменшувати складність заданих функцій у підсумку. Метод також розроблений і застосовується для мінімізації складних булевих функцій. Порівняльний аналіз показав, що запропонований спосіб включення процедури упорядкування оптимальних мінтермів зає змогу отримувати більш точні результати спрощення, використовує менше часу та пам'яті, порівняно з програмою ESPRESSO. Представлена практика упорядкування оптимальних мінтермів для мінімізації булевих функцій.

Наближений синтез – це недавня тенденція логічного синтезу, коли змінюються деякі результати логічної специфікації у межах допустимої похибки заданого додатку, щоб зменшити складність та пришвидшити реалізацію остаточної цифрової компоненти, розглянуто у роботі [14]. Проблема синтезу вирішується за допомогою використання дозволеної гнучкості технології, щоб максимізувати регулярність заданих булевих функцій. Зокрема розглядається два типи регулярності: симетрія та D-скорочуваність. Реалізується два алгоритми з метою знайти, відповідно, симетричне та D-скорочуване наближення заданої функції, у межах допустимої похибки, якщо це можливо. При націлюванні на симетрію технологія характеризує і обчислює найближче симетричне наближення. Представлено поліноміальний евристичний алгоритм для обчислення D-наближення не повністю заданої булевої функції за метрикою бітової помилки. Експериментальні результати на класичних та нових еталонах підтверджують ефективність запропонованих підходів.

Проблема покриття логічної структури та мінімізація PLA, розглядається у роботі [15]. Вирішення проблеми мінімального покриття здійснюється за допомогою, так названого, неявного методу перерахування. Зазначений метод у роботі [15] є модифікацією методу Куайна-МакКласкі, пристосований до комп’ютерної обробки. Метод має розширення, які використовують деякі нові властивості алгоритмів мінімального покриття для прискорення процедури. Для вирішення масштабних задач представлений евристичний алгоритм. Його застосування для мінімізації програмованих логічних масивів (PLA) показаний як приклад. Обговорюється обчислювальний досвід, який представлений для підтвердження покращень задач мінімального покриття методом неявного перерахування.

Програмна реалізація методу Куайна-Мак-Класкі для мінімізації булевих виразів розглядається у роботі [16]. Метод Куайна-МакКласкі (QM) є одним з найпотужніших прийомів для спрощення булевих виразів. У порівнянні з іншими методами, метод QM може обробляти логічні функції з більшим числом змінних. Крім того, метод QM легше впроваджувати до комп’ютерних програм, що робить його ефективною технікою. Для алгоритму QM мінімізації булевих функцій представлений код мовою С.

Новий швидкий метод мінімізації булевих функцій представлений у роботі [17]. Метод використовує нейронні мережі (FNN), що реалізовані у частотній області. На мережу покладається виконання крос-кореляції в частотній області. Доведено розрахунками та підтверджено на практиці, що кількість етапів обчислення, необхідних для мереж FNN, становить менше, ніж це необхідно стандартним нейронним мережам (CNN). Результати моделювання за допомогою MATLAB підтверджують теоретичні обчислення.

Розширений QMC алгоритм (*e* QMC), який підвищує продуктивність процесу мінімізації булевої функції методом Квайна – Мак-Класкі, представлений у роботі [18]. Тут подана демонстрація збільшення швидкості та продуктивності роботи пам'яті комп’ютера шляхом моделювання процесу мінімізації булевих функцій.

За останні кілька десятиліть у галузі методології QCA було витрачено багато зусиль для розроблення ефективних алгоритмів мінімізації булевих функцій, з метою отримати точний, а головне повний список мінімальних простих імплікант. Оскільки складність методу зростає у геометричній прогресії з кожним новим станом, необхідна пам’ять комп’ютера минає поточні комп'ютерні ресурси та поліноміальний час, необхідний для вирішення цієї проблеми. Стаття [19] представляє нову альтернативу існуючим неполіноміальним спробам що повністю вирішує проблему пам'яті. Попередні тести показують, що задача вирішується у сотні разів швидше порівняно з методом eQMC. Хоча швидкість на даний момент не є великою проблемою (eQMC алгоритм досить швидкий для простих даних), це може виявитись важливий при подальшому розвитку до всіх можливих тимчасових порядків або пошуку конфігурацій у даних панелі з часом, у поєднанні з/або автоматичним виявленням складних фактів процедури, тощо.

Розглянуті літературні джерела [13–19] в основному представляють складні алгоритми мінімізації булевих функцій, які ґрунтуються на методі Квайна – Мак-Класкі, його модифікаціях та кубічній техніці. Компенсацією за складність пошуку оптимальної функції для таких алгоритмів може бути наближений синтез – тенденція логічного синтезу, коли змінюються деякі результати логічної специфікації у межах допустимої неоптимальності цифрової схеми, що проектується. Регулярним технологічним пунктом для подібних алгоритмів є порівняння отриманого результату з результатом мінімізації евристичним алгоритмом ESPRESSO. Хоча ESPRESSO не гарантує, що результатом мінімізації буде глобальний мінімум, на практиці він дуже близький і в той же час рішення завжди позбавлене надмірності [20].

Візуально-матрична форма аналітичного методу, що ґрунтується на бінарних комбінаторних системах з повторенням 2-(n, b)-design, 2-(n, x/b)-design, як дещо окреме від складних алгоритмів Куайна-Мак-Класкі та кубічної техніки, не передбачає наближеного результату мінімізації та не виключає ручного способу мінімізації булевих функцій. Алгоритми для автоматизації процедури спрощення логічних функцій аналітичним методом представлено у [12].

Таким чином, складні алгоритми Куайна–Мак-Класкі, його модифікації, кубічна техніка, створені програмні засоби до них, що охоплюють загальну процедуру мінімізації логічних функцій [13–19] та візуально-матрична форма аналітичного методу, посідають відмінні підходи (принципи мінімізації). А відтак вбачають різні перспективи стосовно можливості алгоритмічної мінімізації логічних функцій.

Складові технології, як то «наближений синтез», «евристичний алгоритм», «пристосування до комп'ютерної обробки» є технологічним інструментарієм для поточної практики, який змінюється. Це є підставою вважати, що програмно-технологічна база, яка представлена алгоритмами Куайна-Мак-Класкі, його модифікаціями та кубічною технікою [13–19], є недостатньою (перебуває у стані зміни), для проведення теоретичних досліджень стосовно оптимальної мінімізації булевих функцій. Це визначає необхідність здійснення досліджень з візуально-матричною формою аналітичного методу мінімізації логічних функцій. Зокрема з особливостями порівняно складних алгоритмів проведення логічних операцій поглинання та супер-склеювання змінних, стеком логічних операцій для першої бінарної матриці заданої функції, способами спрощення булевими функціями загального вигляду.

У прикладному відношенні зазначений підхід дозволить розширити можливості технології проектування цифрових компонентів. Хоча алгоритм Куайна–Мак-Класкі добре підходить для реалізації комп'ютерною програмою, результат все ще є неефективним за часом обчислення та використаної пам'яті. Добавлення змінної до функції приблизно подвоює їх обидві, оскільки розмір таблиці істинності збільшується експоненціально з ростом кількості змінних [20]. При збільшенні розрядності булевих функцій масив даних не придатних для спрощення збільшується, а системи 2-(*n*, *b*)-design, 2-(*n*, *x/b*)-design у бінарній конфігурації функції розміщуватись рідше (приклад 12). Чим більша розрядність логічних функцій, тим менша ефективність алгоритму Куайна–Мак-Класкі. У свою чергу, візуально-матрична процедура насамперед знаходить системи 2-(*n*, *b*)-design, 2-(*n*, *x/b*)-design, після чого здійснюється мінімізація. Тому при збільшенні розрядності логічних функцій ефективність візуально-матричної процедури не зменшується.

3. Мета і завдання дослідження

Метою дослідження є встановлення оновленого порядку чергування протоколів рівносильних перетворень для першої та другої матриці заданої логічної функції, порівняно з тим, що розглянуто у [12]. Це дасть змогу визначити новий стандарт процедури, що забезпечить належну ефективність до раніш не врахованих варіантів спрощення булевих функцій аналітичним методом, зокрема у класі ДДНФ та ДКНФ представленні.

Для досягнення мети були поставлені такі завдання:

– встановити особливості застосування логічних операцій поглинання та супер-склеювання змінних для мінімізації булевих функцій аналітичним методом;

– визначити стек логічних операцій аналітичного методу для мінімізації у межах першої та другої матриці заданої логічної функції;

– провести аналіз методів спрощення булевих функцій загального вигляду;

– продемонструвати приклади мінімізацій булевих функцій з метою оцінки ефективності візуально-матричної форми аналітичного методу при мінімізації логічних функцій.

4. Особливості застосування логічних операцій поглинання та суперсклеювання змінних

Логічні операції зі змінними на бінарних і алгебричних структурах функцій, до певної міри, будемо виділяти кольором. Це забезпечить кращу дидактику методу.

Для аналітичного методу мінімізації булевих функцій поглинання змінних з дублюванням елементарних кон'юнкцій може мати такий вигляд:

$$
\begin{vmatrix} 1 \\ 2 \\ 3 \\ 4 \\ 5 \end{vmatrix}
\begin{matrix} 1 & & & 1 & \\ 1 & & & & 0 \\ 1 & 1 & & & \\ 1 & & 0 & & \\ & 0 & 0 & & 1 \end{matrix}
\Bigg| =
\begin{vmatrix} 1 & & & 1 & \\ 1 & & & & 0 \\ 1 & 1 & & & \\ 1 & & 0 & 0 & \\ 1 & & 0 & 1 & \\ & 0 & 0 & & 1 \end{vmatrix} =
\begin{vmatrix} 1 & & & 1 & \\ 1 & & & & 0 \\ 1 & 1 & & & \\ 1 & & 0 & 0 & \\ & 0 & 0 & & 1 \end{vmatrix} =
$$

$$
= \begin{vmatrix} 1 & & & 1 & \\ 1 & & & & 0 \\ 1 & 1 & & & \\ 1 & & 0 & 0 & 0 \\ 1 & & 0 & 0 & 1 \\ & 0 & 0 & & 1 \end{vmatrix} =
\begin{vmatrix} 1 & & & 1 & \\ 1 & & & & 0 \\ 1 & 1 & & & \\ 1 & & 0 & 0 & 1 \\ & 0 & 0 & & 1 \end{vmatrix} =
\begin{vmatrix} 1 & & & 1 & \\ 1 & & & & 0 \\ 1 & 1 & & & \\ 1 & 0 & 0 & 0 & 1 \\ 1 & 1 & 0 & 0 & 1 \\ & 0 & 0 & & 1 \end{vmatrix} =
$$

$$
= \begin{vmatrix} 1 & & & 1 & \\ 1 & & & & 0 \\ 1 & 1 & & & \\ & 0 & 0 & & 1 \end{vmatrix}.
$$

$$
x_1x_4 + x_1\overline{x}_5 + x_1x_2 + x_1\overline{x}_3 + \overline{x}_2\overline{x}_3x_5 = x_1x_4 + x_1\overline{x}_5 + x_1x_2 + x_1\overline{x}_3\overline{x}_4 + x_1\overline{x}_3x_4 + \overline{x}_2\overline{x}_3x_5 =
$$

$$
= x_1x_4 + x_1\overline{x}_5 + x_1x_2 + x_1\overline{x}_3\overline{x}_4 + \overline{x}_2\overline{x}_3x_5 = x_1x_4 + x_1\overline{x}_5 + x_1x_2 + x_1\overline{x}_3\overline{x}_4\overline{x}_5 +
$$

$$
+ x_1\overline{x}_3\overline{x}_4x_5 + \overline{x}_2\overline{x}_3x_5 = x_1x_4 + x_1\overline{x}_5 + x_1x_2 + x_1\overline{x}_3\overline{x}_4x_5 + \overline{x}_2\overline{x}_3x_5 = x_1x_4 + x_1\overline{x}_5 + x_1x_2 +
$$

$$
+ x_1\overline{x}_2\overline{x}_3\overline{x}_4x_5 + x_1x_2\overline{x}_3\overline{x}_4x_5 + \overline{x}_2\overline{x}_3x_5 = x_1x_4 + x_1\overline{x}_5 + x_1x_2 + \overline{x}_2\overline{x}_3x_5.
$$

Дублювання конституант з наступною логічною операцією простого склеювання змінних може мати такий, наприклад, вигляд:

$$
\begin{vmatrix} 0 & & 0 \\ 0 & 1 & 1 \\ 1 & 0 & 0 \\ 1 & & 1 \end{vmatrix} = \begin{vmatrix} 0 & 0 & 0 \\ 0 & 1 & 0 \\ 0 & 1 & 1 \\ 1 & 0 & 0 \\ 1 & & 1 \end{vmatrix} = \begin{vmatrix} 0 & 1 & \\ & 0 & 0 \\ 1 & & 1 \end{vmatrix}. \qquad (4)
$$

Отримати інший мінімальний вираз стає можливим при дублюванні конституанти для нижнього рядка першої бінарної матриці (4).

Дублювання конституант з наступною операцією супер-склеювання змінних може мати такий, наприклад, вигляд:

$$
\begin{vmatrix} 1 & 1 & 0 & & 1 & 0 \\ 1 & 1 & 1 & & 0 & 0 \\ 1 & 1 & 0 & 1 & 0 & 0 \\ 1 & 1 & 0 & 1 & 0 & 1 \\ 1 & 1 & 0 & 1 & 1 & 1 \\ 1 & 1 & 1 & 0 & 0 & 1 \\ 1 & 1 & 1 & 0 & 1 & 0 \\ 1 & 1 & 1 & 0 & 1 & 1 \end{vmatrix} = \begin{vmatrix} 1 & 1 & 0 & & 1 & 0 \\ 1 & 1 & 1 & & 0 & 0 \\ 1 & 1 & 0 & 1 & 0 & 0 \\ 1 & 1 & 0 & 1 & 0 & 1 \\ 1 & 1 & 0 & 1 & 1 & 1 \\ 1 & 1 & 0 & 1 & 1 & 0 \\ 1 & 1 & 1 & 0 & 0 & 1 \\ 1 & 1 & 1 & 0 & 1 & 0 \\ 1 & 1 & 1 & 0 & 1 & 1 \\ 1 & 1 & 1 & 0 & 0 & 0 \end{vmatrix} = \begin{vmatrix} 1 & 1 & 0 & & 1 & 0 \\ 1 & 1 & 1 & & 0 & 0 \\ 1 & 1 & 0 & 1 & & \\ 1 & 1 & 1 & 0 & & \end{vmatrix}. \qquad (5)
$$

$$
\begin{aligned}
& x_1x_2\overline{x}_3x_5\overline{x}_6 + x_1x_2x_3\overline{x}_5\overline{x}_6 + x_1x_2\overline{x}_3x_4\overline{x}_5\overline{x}_6 + x_1x_2\overline{x}_3x_4\overline{x}_5x_6 + x_1x_2\overline{x}_3x_4x_5x_6 + \\
& + x_1x_2x_3\overline{x}_4\overline{x}_5x_6 + x_1x_2x_3\overline{x}_4x_5\overline{x}_6 + x_1x_2x_3\overline{x}_4x_5x_6 = x_1x_2\overline{x}_3x_5\overline{x}_6 + x_1x_2x_3\overline{x}_5\overline{x}_6 + \\
& + x_1x_2\overline{x}_3x_4\overline{x}_5\overline{x}_6 + x_1x_2\overline{x}_3x_4\overline{x}_5x_6 + x_1x_2\overline{x}_3x_4x_5x_6 + x_1x_2\overline{x}_3x_4x_5\overline{x}_6 + \\
& + x_1x_2x_3\overline{x}_4\overline{x}_5x_6 + x_1x_2x_3\overline{x}_4x_5\overline{x}_6 + x_1x_2x_3\overline{x}_4x_5x_6 + x_1x_2x_3\overline{x}_4\overline{x}_5\overline{x}_6 = x_1x_2\overline{x}_3x_5\overline{x}_6 + \\
& + x_1x_2x_3\overline{x}_5\overline{x}_6 + x_1x_2\overline{x}_3x_4\left(\overline{x}_5\overline{x}_6 + \overline{x}_5x_6 + x_5x_6 + x_5\overline{x}_6\right) + \\
& + x_1x_2x_3\overline{x}_4\left(\overline{x}_5x_6 + x_5\overline{x}_6 + x_5x_6 + \overline{x}_5\overline{x}_6\right) = \\
& = x_1x_2\overline{x}_3x_5\overline{x}_6 + x_1x_2x_3\overline{x}_5\overline{x}_6 + x_1x_2\overline{x}_3x_4 + x_1x_2x_3\overline{x}_4.
\end{aligned} \qquad (6)
$$

Результати матричного (5) та алгебричного (6) способу мінімізації логічного виразу співпадають.

Представлені порівняно складніші алгоритми застосування логічних операцій поглинання та супер-склеювання змінних розширюють варіанти їхнього використання, що дає збільшення ефективності процедури мінімізації булевих функцій аналітичним методом.

5. Стек логічних операцій першої та другої бінарних матриць аналітичного методу

Ефективність мінімізації булевих функцій аналітичним методом залежить від комбінування послідовності логічних операцій з використанням різних способів склеювання змінних – простого та супер-склеювання у першій, а, в окремих випадках, і у другій бінарній матриці. Для оцінки результату перетворень у першій матриці, за обраним стеком (переліком) логічних операцій, можна встановити критерій ефективності перетворення логічних виразів. Оцінювати результат перетворення першої матриці можна за кількістю термів, що залишились. При рівності кількості термів, критерієм може бути кількість літералів. При рівності кількості термів і літералів, критерієм може бути кількість інвертованих змінних.

Приклад 2. Обрати оптимальний стек логічних операцій у першій матриці 4-змінної булевої функції з восьми наборів, представленої у ДДНФ.

Перший стек логічних операцій: просте склеювання змінних.

$$\begin{vmatrix} 0 & 0 & 1 & 1 \\ 0 & 1 & 0 & 1 \\ 1 & 0 & 0 & 0 \\ 1 & 0 & 0 & 1 \\ 1 & 0 & 1 & 1 \\ 1 & 1 & 0 & 1 \\ 1 & 1 & 1 & 0 \\ 1 & 1 & 1 & 1 \end{vmatrix} = \begin{vmatrix} & 0 & 1 & 1 \\ & 1 & 0 & 1 \\ 1 & 0 & 0 & \\ 1 & 1 & 1 & \end{vmatrix}$$

Другий стек логічних операцій: супер-склеювання змінних.

$$\begin{vmatrix} 0 & 0 & 1 & 1 \\ 0 & 1 & 0 & 1 \\ 1 & 0 & 0 & 0 \\ 1 & 0 & 0 & 1 \\ 1 & 0 & 1 & 1 \\ 1 & 1 & 0 & 1 \\ 1 & 1 & 1 & 0 \\ 1 & 1 & 1 & 1 \end{vmatrix} = \begin{vmatrix} 0 & 0 & 1 & 1 \\ 0 & 1 & 0 & 1 \\ 1 & 0 & 0 & 0 \\ 1 & & & 1 \\ 1 & 1 & 1 & 0 \end{vmatrix} = \begin{vmatrix} & 0 & 1 & 1 \\ & 1 & 0 & 1 \\ 1 & 0 & 0 & \\ 1 & & & 1 \\ 1 & 1 & 1 & \end{vmatrix} =$$

$$= \begin{vmatrix} & 0 & 1 & 1 \\ & 1 & 0 & 1 \\ 1 & 0 & 0 & \\ 1 & 0 & 0 & 1 \\ 1 & 1 & 0 & 1 \\ 1 & 0 & 1 & 1 \\ 1 & 1 & 1 & 1 \\ 1 & 1 & 1 & \end{vmatrix} = \begin{vmatrix} & 0 & 1 & 1 \\ & 1 & 0 & 1 \\ 1 & 0 & 0 & \\ 1 & 1 & 1 & \end{vmatrix}$$

Два стеки логічних операцій дають однаковий результат мінімізації булевої функції. Однак результатом перетворення першого стеку є чотири терми (переписуються у другу матрицю), а результатом перетворення другого стеку є п’ять термів. Тому оптимальним, згідно з критерієм, буде перетворення за першим стеком логічних операцій.

Розглянута бінарна конфігурація булевої функції є рідким випадком, коли для мінімізації функції оптимальними є логічні операції простого склеювання змінних. У переважній більшості випадків оптимальні перетворення у першій матриці здійснюються за допомогою операції супер-склеювання змінних [12].

Приклад 3. Мінімізувати логічну функцію F (x_1, x_2, x_3, x_4) аналітичним методом, яку представимо у канонічній формі [13]:

$$F = \Sigma\ (4, 7, 9, 10, 12, 13, 14, 15). \qquad (7)$$

Примітка: значення в Σ є мінтерми для рядків таблиці істинності, коли функція F (x_1, x_2, x_3, x_4) повертає «1» на виході.

Провівши аналіз бінарної конфігурації таблиці істинності функції F (x_1, x_2, x_3, x_4) (7), приймаємо рішення, що оптимальним стеком логічних операцій для рівносильних перетворень у першій матриці будуть операції простого склеювання змінних. Тоді спрощення функції $F(x_1, x_2, x_3, x_4)$ (7) [13] аналітичним методом набуває такого оптимального вигляду:

$$F\left(x_1,x_2,x_3,x_4\right)=\begin{array}{|c|cccc|}4 & 0 & 1 & 0 & 0\\ 7 & 0 & 1 & 1 & 1\\ 9 & 1 & 0 & 0 & 1\\ 10 & 1 & 0 & 1 & 0\\ 12 & 1 & 1 & 0 & 0\\ 13 & 1 & 1 & 0 & 1\\ 14 & 1 & 1 & 1 & 0\\ 15 & 1 & 1 & 1 & 1\end{array}=\begin{vmatrix} & 1 & 0 & 0\\ & 1 & 1 & 1\\ 1 & & 0 & 1\\ 1 & & 1 & 0\end{vmatrix}=$$
$$=x_2\overline{x_3}\,\overline{x_4}+x_2x_3x_4+x_1\overline{x_3}x_4+x_1x_3\overline{x_4}. \tag{8}$$

Результат мінімізації (8) функції $F(x_1, x_2, x_3, x_4)$ (7) співпадає з результатом мінімізації методом оптимального упорядкування мінтермів [13], однак процедура мінімізації аналітичним методом простіша.

Приклад 4. Обрати оптимальний стек логічних операцій для спрощення булевої функції, $F(x_1, x_2, x_3)$ [21] представленої в алгебричній формі.

$$f\left(x_1,x_2,x_3\right)=\overline{x_1}\,\overline{x_2}\,\overline{x_3}+x_1\overline{x_2}\,\overline{x_3}+\overline{x_1}x_2\overline{x_3}+x_1\overline{x_2}x_3+\overline{x_1}x_2x_3+x_1x_2x_3. \tag{9}$$

Рішення:

Для спрощення функції $f(x_1, x_2, x_3)$ (9) застосуємо дублювання конституант з наступною операцією простого склеювання змінних (п. 4):

$$f\left(x_1,x_2,x_3\right)=\begin{vmatrix}0 & 0 & 0\\ 1 & 0 & 0\\ 0 & 1 & 0\\ 1 & 0 & 1\\ 0 & 1 & 1\\ 1 & 1 & 1\end{vmatrix}=\begin{vmatrix}0 & 0 & 0\\ 0 & 1 & 0\\ 0 & 1 & 1\\ 1 & 0 & 0\\ 1 & 0 & 1\\ 1 & 1 & 1\end{vmatrix}=\begin{vmatrix}0 & & 0\\ 0 & 1 & 1\\ 1 & 0 & 0\\ 1 & & 1\end{vmatrix}=$$
$$=\begin{vmatrix}0 & 0 & 0\\ 0 & 1 & 0\\ 0 & 1 & 1\\ 1 & 0 & 0\\ 1 & & 1\end{vmatrix}=\begin{vmatrix}0 & 1 & \\ & 0 & 0\\ 1 & & 1\end{vmatrix} \tag{10}$$

Набори змінних у першій матриці (10) викладаємо у лексікографічному порядку. Таке упорядкування потрібно робити завжди при використанні візуально-матричної форми аналітичного методу. Результат упорядкування переписується до другої матриці (10). У другій матриці (10) проводимо дві операції простого склеювання змінних, після чого результат склеювання змінних переписуємо до третьої матриці (10). Далі застосовуємо дублювання конституант (виділено червоним кольором). У четвертій матриці проводимо дві операції простого склеювання змінних. У підсумку отримуємо мінімальну функцію:

$$f\left(x_1, x_2, x_3\right) = \overline{x}_1 x_2 + \overline{x}_2\,\overline{x}_3 + x_1 x_3. \tag{11}$$

Результат мінімізації (11) функції $F(x_1, x_2, x_3)$ (9) співпадає з результатом мінімізації за допомогою матриці відстаней [21]. Для розглянутого прикладу процедура мінімізації аналітичним методом є простішою.

Приклад 5. Обрати оптимальний стек логічних операцій для рівносильних перетворень у першій та у другій бінарних матрицях булевої функції, $F(x_1, x_2, x_3, x_4, x_5)$ (рис. 1) [12] представленої у ДДНФ.

Ефективність мінімізації функції $F(x_1, x_2, x_3, x_4, x_5)$ (рис. 1) ґрунтується на першочерговому застосуванні логічної операції супер-склеювання змінних у першій матриці. Супер-склеювання змінних можливе для блоків: 1, 3, 5, 7 (виділено зеленим кольором), 4, 12, 20, 28 (виділено червоним кольором), 14, 15, 30, 31 (виділено синім кольором). Зазначені блоки представляють собою повну бінарну комбінаторну систему з повторенням 2-(2, 4)-design [9]. Логічна операція простого склеювання змінних у першій матриці не застосовується. Результат мінімізації у першій матриці за допомогою логічних операцій супер-склеювання змінних переписується до другої матриці.

Мінімізація функції у другій матриці також здійснюється за допомогою супер-склеювання змінних, однак для утворення систем 2-(2, 4)-design потрібно застосувати дублювання відповідних конституант (п. 4). Результат дублювання конституант переписується до третьої матриці. Далі мінімізація функції проводиться за допомогою простого склеювання змінних, дублювання елементарних кон’юнкцій та операцій поглинання змінних.

Результат мінімізації функції $F(x_1, x_2, x_3, x_4, x_5)$ (рис. 1) співпадає з результатом мінімізації у [12]. У кожному випадку мінімізація функції $F(x_1, x_2, x_3, x_4, x_5)$ (рис. 1) проведена ефективно, хоча процедури мінімізації використовують різні протоколи.

При комбінуванні послідовності логічних операцій з використанням різних способів склеювання змінних – простого та супер-склеювання існує невелике число випадків, коли мінімізація функції є більш ефективна, якщо у першій матриці спочатку застосувати операцію простого склеювання змінних [12]. Таким чином, необхідний короткий аналіз для першочергового застосування операцій у першій бінарній матриці (встановити оптимальний стек логічних операцій). Це забезпечує належну ефективність мінімізації до раніш не врахованих варіантів спрощення булевих функцій візуально-матричною формою аналітичного методу [12]. Для ряду випадків вибір оптимального стеку потрібний і для обчислень у другій бінарній матриці.

$$\begin{vmatrix} 1 \\ 2 \\ 3 \\ 4 \\ 5 \\ 7 \\ 9 \\ 11 \\ 12 \\ 13 \\ 14 \\ 15 \\ 16 \\ 17 \\ 18 \\ 20 \\ 22 \\ 26 \\ 28 \\ 30 \\ 31 \end{vmatrix}\begin{vmatrix} 0 & 0 & 0 & 0 & 1 \\ 0 & 0 & 0 & 1 & 0 \\ 0 & 0 & 0 & 1 & 1 \\ 0 & 0 & 1 & 0 & 0 \\ 0 & 0 & 1 & 0 & 1 \\ 0 & 0 & 1 & 1 & 1 \\ 0 & 1 & 0 & 0 & 1 \\ 0 & 1 & 0 & 1 & 1 \\ 0 & 1 & 1 & 0 & 0 \\ 0 & 1 & 1 & 0 & 1 \\ 0 & 1 & 1 & 1 & 0 \\ 0 & 1 & 1 & 1 & 1 \\ 1 & 0 & 0 & 0 & 0 \\ 1 & 0 & 0 & 0 & 1 \\ 1 & 0 & 0 & 1 & 0 \\ 1 & 0 & 1 & 0 & 0 \\ 1 & 0 & 1 & 1 & 0 \\ 1 & 1 & 0 & 1 & 0 \\ 1 & 1 & 1 & 0 & 0 \\ 1 & 1 & 1 & 1 & 0 \\ 1 & 1 & 1 & 1 & 1 \end{vmatrix} = \begin{vmatrix} 0 & 0 & & & 1 \\ 0 & 0 & 0 & 1 & 0 \\ 0 & 1 & 0 & 0 & 1 \\ 0 & 1 & 0 & 1 & 1 \\ 0 & 1 & 1 & 0 & 1 \\ 1 & 0 & 0 & 0 & 0 \\ 1 & 0 & 0 & 0 & 1 \\ 1 & 0 & 0 & 1 & 0 \\ 1 & 0 & 1 & 1 & 0 \\ 1 & 1 & 0 & 1 & 0 \\ & & 1 & 0 & 0 \\ & 1 & 1 & 1 & \end{vmatrix} = \begin{vmatrix} 0 & 0 & & & 1 \\ 0 & 0 & 0 & 1 & 0 \\ 0 & 1 & 0 & 0 & 1 \\ 0 & 1 & 0 & 1 & 1 \\ 0 & 1 & 1 & 0 & 1 \\ 0 & 1 & 1 & 1 & 1 \\ 1 & 0 & 0 & 0 & 0 \\ 1 & 0 & 0 & 0 & 1 \\ 1 & 0 & 0 & 1 & 0 \\ 1 & 0 & 1 & 0 & 0 \\ 1 & 0 & 1 & 1 & 0 \\ 1 & 1 & 0 & 1 & 0 \\ & & 1 & 0 & 0 \\ & 1 & 1 & 1 & \end{vmatrix} = \begin{vmatrix} 0 & 0 & & & 1 \\ 0 & 0 & 0 & 1 & 0 \\ 0 & 1 & & & 1 \\ 1 & 0 & & & 0 \\ 1 & 0 & 0 & 0 & 1 \\ 1 & 1 & & 1 & 0 \\ & & 1 & 0 & 0 \\ & 1 & 1 & 1 & \end{vmatrix} =$$

$$= \begin{vmatrix} 0 & & & & 1 \\ 0 & 0 & 0 & 1 & \\ 1 & 0 & & & 0 \\ 1 & 0 & 0 & 0 & \\ 1 & & & 1 & 0 \\ & & 1 & 0 & 0 \\ & 1 & 1 & 1 & \end{vmatrix} = \begin{vmatrix} 0 & & & & 1 \\ 0 & 0 & 0 & 1 & \\ 1 & 0 & & 0 & 0 \\ 1 & 0 & & 1 & 0 \\ 1 & 0 & 0 & 0 & \\ 1 & & & 1 & 0 \\ & & 1 & 0 & 0 \\ & 1 & 1 & 1 & \end{vmatrix} = \begin{vmatrix} 0 & & & & 1 \\ 0 & 0 & 0 & 1 & \\ 1 & 0 & 0 & 0 & 0 \\ 1 & 0 & 1 & 0 & 0 \\ 1 & 0 & 0 & 0 & \\ 1 & & & 1 & 0 \\ & & 1 & 0 & 0 \\ & 1 & 1 & 1 & \end{vmatrix} = \begin{vmatrix} 0 & & & & 1 \\ 0 & 0 & 0 & 1 & \\ 1 & 0 & 0 & 0 & \\ 1 & & & 1 & 0 \\ & & 1 & 0 & 0 \\ & 1 & 1 & 1 & \end{vmatrix}.$$

Рис. 1. Спрощення булевої функції за допомогою дублювання конституант та елементарних кон'юнкцій

6. Аналіз методів спрощення булевих функції загального вигляду

Рівносильні перетворення першої матриці переводять ДДНФ заданої функції у довільну ДНФ. Ефективне спрощення довільної ДНФ булевої функції здійснюється за допомогою логічних операцій, розглянутих у п. 4, 5. Не виключено, що довільну ДНФ можна спрощувати методом паралельного розчеплення кон'юнктермів [22]. ДНФ функції, зокрема спрощується за допомогою методу Блейка - Порецького [23].

Приклад 6. Знайти мінімальну алгебричну форму булевої функції f (a, b, c, d, e) (12) [3, 24].

$$f\left(a,b,c,d,e\right)=ab+a\bar{c}d+bc+bde+\bar{a}de+\bar{c}de. \tag{12}$$

Рішення:

$$f\left(a,b,c,d,e\right)=\begin{vmatrix} 1 & 1 & & & \\ 1 & & 0 & 1 & \\ & 1 & 1 & & \\ & 1 & & 1 & 1 \\ 0 & & & 1 & 1 \\ & & 0 & 1 & 1 \end{vmatrix}=\begin{vmatrix} 1 & 1 & & & \\ 1 & & 0 & 1 & \\ & 1 & 1 & & \\ 0 & & & 1 & 1 \\ & & 0 & 1 & 1 \end{vmatrix}=$$

Для дидактичної зручності образних перетворень права матриця переписана до нового рядка, оскільки поточна процедура спрощення використовує спільний блок з попередньої логічної операції:

$$=\begin{vmatrix} 1 & 1 & & & \\ 1 & & 0 & 1 & \\ & 1 & 1 & & \\ 0 & & & 1 & 1 \\ & & 0 & 1 & 1 \end{vmatrix}=\begin{vmatrix} 1 & 1 & & & \\ 1 & & 0 & 1 & \\ & 1 & 1 & & \\ 0 & & & 1 & 1 \end{vmatrix}=b\left(a+c\right)+d\left(a\bar{c}+\bar{a}e\right).$$

Остаточний результат $f\left(a,b,c,d,e\right)=b\left(a+c\right)+d\left(a\bar{c}+\bar{a}e\right)$ містить вісім літералів. Це на один літерал менше ніж у [3] і на два літерала менше ніж у [24].

7. Результати мінімізації булевих функцій візуально-матричною формою аналітичного методу

Протокол з відносно складним алгоритмом застосування операцій поглинання та супер-склеювання змінних (п. 4), стек логічних операцій у першій матриці (п. 5) визначає новий, порівняно з [12], стандарт мінімізації булевих функцій. Це збільшує ефективність процедури, що дає змогу, зокрема, спрощувати логічні функції з відносно більшим числом вхідних змінних ручним способом.

Приклад 7. Мінімізувати логічну функцію F (x_1, x_2, x_3, x_4, x_5) аналітичним методом, яку представимо у канонічній формі [25].

$$F=\Sigma\ (0, 2, 5, 6, 7, 8, 10, 16, 19, 20, 23, 27, 31). \tag{13}$$

Рішення:

$$F\left(x_1, x_2, x_3, x_4, x_5\right) =$$

$$= \left|\begin{array}{c|ccccc} 0 & 0 & 0 & 0 & 0 & 0 \\ 2 & 0 & 0 & 0 & 1 & 0 \\ 5 & 0 & 0 & 1 & 0 & 1 \\ 6 & 0 & 0 & 1 & 1 & 0 \\ 7 & 0 & 0 & 1 & 1 & 1 \\ 8 & 0 & 1 & 0 & 0 & 0 \\ 10 & 0 & 1 & 0 & 1 & 0 \\ 16 & 1 & 0 & 0 & 0 & 0 \\ 19 & 1 & 0 & 0 & 1 & 1 \\ 20 & 1 & 0 & 1 & 0 & 0 \\ 23 & 1 & 0 & 1 & 1 & 1 \\ 27 & 1 & 1 & 0 & 1 & 1 \\ 31 & 1 & 1 & 1 & 1 & 1 \end{array}\right| = \left|\begin{array}{ccccc} 0 & & 0 & & 0 \\ 0 & 0 & 1 & 0 & 1 \\ 0 & 0 & 1 & 1 & \\ 1 & 0 & & 0 & 0 \\ 1 & & & 1 & 1 \end{array}\right| = \left|\begin{array}{ccccc} 0 & & 0 & & 0 \\ 0 & 0 & 1 & & 1 \\ 0 & 0 & 1 & 1 & \\ 1 & 0 & & 0 & 0 \\ 1 & & & 1 & 1 \end{array}\right|. \tag{14}$$

У табл. 1 представлені результати мінімізації функції F (x_1, x_2, x_3, x_4, x_5) (13) за допомогою Карти Карно, методом Квайна – Мак-Класкі, методом невизначених коефіцієнтів, [25] та аналітичним методом.

Таблиця 1
Результат мінімізації функції F (x_1, x_2, x_3, x_4, x_5) (13)

Картою Карно, методом Квайна-Мак-Класкі, методом невизначених коефіцієнтів	Аналітичним методом
$F = \overline{x}_1\overline{x}_2 x_3 x_5 + x_1\overline{x}_2\overline{x}_4\overline{x}_5 + \overline{x}_1\overline{x}_3 x_4\overline{x}_5 +$ $+ \overline{x}_1\overline{x}_2 x_4\overline{x}_5 + \overline{x}_1\overline{x}_3\overline{x}_4\overline{x}_5 + x_1 x_4 x_5$	$F = \overline{x}_1\overline{x}_3\overline{x}_5 + \overline{x}_1\overline{x}_2 x_3 x_5 + \overline{x}_1\overline{x}_2 x_3 x_4 +$ $+ x_1\overline{x}_2\overline{x}_4\overline{x}_5 + x_1 x_4 x_5$

Споглядаючи табл. 1, бачимо, що результатом мінімізації функції F (x_1, x_2, x_3, x_4, x_5) (14) аналітичним методом є мінімальна функція, що містить п’ять мінтермів. Це на один мінтерм менше ніж у [25].

Результат мінімізації функції F (x_1, x_2, x_3, x_4, x_5) (13) за допомогою програмного забезпечення Logic Friday 1.1.4 [26] є такий:

$$F=A'B'DE'+A'C'E'+ADE+AB'D'E'+A'B'CE. \tag{15}$$

Мінімальна функція (15) вміщує 11 інвертованих змінних. Це на одну інвертовану змінну більше ніж у (14).

Logic Friday надає графічний інтерфейс для програми Espresso [27].

Приклад 8. Мінімізувати логічну функцію F ($x_1, x_2, x_3, x_4, x_5, x_6$) аналітичним методом, яку представимо у канонічній формі [28].

$$F=\sum\begin{pmatrix} 0,\ 1,\ 2,\ 3,\ 4,\ 9,\ 10,\ 11, 12,\ 15,\ 18,\ 19, 20,\ 21,\ 22,\ 23, \\ 26,\ 27,\ 28,\ 29,\ 32, 33,\ 38,\ 39,\ 40,\ 41,\ 42,\ 43,\ 46, \\ 47,\ 48,\ 51,\ 52,\ 53,\ 54,\ 55,\ 56,\ 57,\ 62,\ 63 \end{pmatrix}. \qquad (16)$$

Рішення:

$$F\left(x_1,x_2,x_3,x_4,x_5,x_6\right)=$$

$$=\left|\begin{array}{r|cccccc||r|cccccc}
0 & 0&0&0&0&0&0 & 32 & 1&0&0&0&0&0\\
1 & 0&0&0&0&0&1 & 33 & 1&0&0&0&0&1\\
2 & 0&0&0&0&1&0 & 38 & 1&0&0&1&1&0\\
3 & 0&0&0&0&1&1 & 39 & 1&0&0&1&1&1\\
4 & 0&0&0&1&0&0 & 40 & 1&0&1&0&0&0\\
9 & 0&0&1&0&0&1 & 41 & 1&0&1&0&0&1\\
10 & 0&0&1&0&1&0 & 42 & 1&0&1&0&1&0\\
11 & 0&0&1&0&1&1 & 43 & 1&0&1&0&1&1\\
12 & 0&0&1&1&0&0 & 46 & 1&0&1&1&1&0\\
15 & 0&0&1&1&1&1 & 47 & 1&0&1&1&1&1\\
18 & 0&1&0&0&1&0 & 48 & 1&1&0&0&0&0\\
19 & 0&1&0&0&1&1 & 51 & 1&1&0&0&1&1\\
20 & 0&1&0&1&0&0 & 52 & 1&1&0&1&0&0\\
21 & 0&1&0&1&0&1 & 53 & 1&1&0&1&0&1\\
22 & 0&1&0&1&1&0 & 54 & 1&1&0&1&1&0\\
23 & 0&1&0&1&1&1 & 55 & 1&1&0&1&1&1\\
26 & 0&1&1&0&1&0 & 56 & 1&1&1&0&0&0\\
27 & 0&1&1&0&1&1 & 57 & 1&1&1&0&0&1\\
28 & 0&1&1&1&0&0 & 62 & 1&1&1&1&1&0\\
29 & 0&1&1&1&0&1 & 63 & 1&1&1&1&1&1
\end{array}\right|=
\left|\begin{array}{cccccc}
0&0&0&0&&\\
0&0&&1&0&0\\
0&0&1&0&0&1\\
0&0&1&0&1&0\\
0&0&1&&1&1\\
0&1&&0&1&\\
&1&0&1&&\\
0&1&1&1&0&\\
1&0&0&0&0&\\
1&0&&1&1&\\
1&0&1&0&&\\
1&1&0&0&0&0\\
1&1&0&0&1&1\\
1&1&1&0&0&\\
1&1&1&1&1&
\end{array}\right|=
\left|\begin{array}{cccccc}
0&0&0&0&&\\
0&&&1&0&0\\
0&0&1&0&&1\\
0&0&1&0&1&\\
0&0&1&&1&1\\
0&1&&0&1&\\
&1&0&1&&\\
0&1&&1&0&\\
&0&0&0&0&\\
1&0&&1&1&\\
1&0&1&0&&\\
1&&0&0&0&0\\
&1&0&0&1&1\\
1&&1&0&0&\\
1&1&&1&1&
\end{array}\right|=$$

$$=\left|\begin{array}{cccccc}
0&0&0&0&&\\
0&&&1&0&0\\
&0&1&0&&1\\
&0&1&0&1&\\
0&0&1&&1&1\\
0&1&&0&1&\\
&1&0&1&&\\
0&1&&1&0&\\
&0&0&0&0&\\
1&&&1&1&\\
1&0&1&0&&\\
1&&&0&0&0\\
&1&0&&1&1\\
1&&1&0&0&
\end{array}\right|=
\left|\begin{array}{cccccc}
0&0&0&0&1&\\
0&&&1&0&0\\
&0&1&0&&1\\
&0&1&0&1&\\
0&0&1&&1&1\\
0&1&&0&1&\\
&1&0&1&&\\
0&1&&1&0&\\
&0&0&0&0&\\
1&&&1&1&\\
1&&&0&0&0\\
&1&0&&1&1\\
1&&1&0&0&
\end{array}\right|=
\left|\begin{array}{cccccc}
0&0&&0&1&\\
0&&&1&0&0\\
&0&1&0&&1\\
&0&1&0&1&\\
0&0&1&0&1&1\\
0&0&1&1&1&1\\
0&1&&0&1&\\
&1&0&1&&\\
0&1&&1&0&\\
&0&0&0&0&\\
1&&&1&1&\\
1&&&0&0&0\\
&1&0&&1&1\\
1&&1&0&0&
\end{array}\right|=
\left|\begin{array}{cccccc}
0&&&1&0&0\\
&0&1&0&&1\\
&0&1&0&1&\\
&0&1&&1&1\\
0&&&0&1&\\
&1&0&1&&\\
0&1&&1&0&\\
&0&0&0&0&\\
1&&&1&1&\\
1&&&0&0&0\\
&1&0&&1&1\\
1&&1&0&0&
\end{array}\right|. \qquad (17)$$

У табл. 2 представлені результати мінімізації функції F (x_1, x_2, x_3, x_4, x_5, x_6) (16) за допомогою програмного забезпечення, що використовує файли ASCII для імпорту і експорту даних та ґрунтується на методі Квайна – Мак-Класкі [28] і аналітичного методу.

З огляду табл. 2 видно, що результатом мінімізації функції F (x_1, x_2, x_3, x_4, x_5, x_6) (16) аналітичним методом є мінімальна функція, що містить двадцять три інвертованих змінних. Це на одну інвертовану змінну менше ніж у [28].

Результати мінімізації функції F (x_1, x_2, x_3, x_4, x_5, x_6) (16) за допомогою програмного забезпечення JQM – Java Quine McChuskey 1.2.4 [29] та аналітичного методу співпадають.

Таблиця 2

Результат мінімізації функції F (x_1, x_2, x_3, x_4, x_5, x_6) (16)

Програмним забезпечення, що ґрунтується на методі Квайна-Мак-Класкі [28]	Аналітичним методом
$F = \bar{x}_1 x_4 \bar{x}_5 \bar{x}_6 + \bar{x}_2 x_3 x_5 x_6 + \bar{x}_1 x_2 x_4 \bar{x}_5 +$ $+ \bar{x}_1 \bar{x}_2 \bar{x}_4 x_6 + \bar{x}_1 \bar{x}_4 x_5 + x_2 \bar{x}_3 x_4 +$ $+ \bar{x}_2 x_3 \bar{x}_4 x_5 + \bar{x}_2 \bar{x}_3 \bar{x}_4 \bar{x}_5 + x_1 x_4 x_5 +$ $+ x_1 \bar{x}_4 \bar{x}_5 \bar{x}_6 + x_2 \bar{x}_3 x_5 x_6 + x_1 x_3 \bar{x}_4 \bar{x}_5$	$F = \bar{x}_1 x_4 \bar{x}_5 \bar{x}_6 + \bar{x}_2 x_3 \bar{x}_4 x_6 + \bar{x}_2 x_3 \bar{x}_4 x_5 +$ $+ \bar{x}_2 x_3 x_5 x_6 + \bar{x}_1 \bar{x}_4 x_5 + x_2 \bar{x}_3 x_4 +$ $+ \bar{x}_1 x_2 x_4 \bar{x}_5 + \bar{x}_2 \bar{x}_3 \bar{x}_4 \bar{x}_5 + x_1 x_4 x_5 +$ $+ x_1 \bar{x}_4 \bar{x}_5 \bar{x}_6 + x_2 \bar{x}_3 x_5 x_6 + x_1 x_3 \bar{x}_4 \bar{x}_5$

Результат мінімізації функції F (x_1, x_2, x_3, x_4, x_5, x_6) (16) за допомогою програмного забезпечення Logic Friday 1.1.4 [26] є такий:

$$F=AD'E'F'+B'CD'E+B'D'E'F+B'C'D'E'+A'DE'F'+A'BDE'+$$
$$+ACD'E'+BC'D+ADE+B'CEF+A'D'E+BC'EF. \tag{18}$$

Мінімальна функція (18) вміщує 24 інвертованих змінних. Це на одну інвертовану змінну більше ніж у (17).

Приклад 9. Мінімізувати логічну функцію F (x_1, x_2, x_3, x_4, x_5, x_6) аналітичним методом, яка задана у ДНФ (табл. 3) [25].

Рішення:

$$F\left(x_1,x_2,x_3,x_4,x_5,x_6\right)=$$

$$=\begin{vmatrix}
0&0&1&&0&1\\
0&1&0&&1&0\\
0&1&0&&1&1\\
0&1&1&&0&1\\
1&0&0&&1&1\\
1&0&1&&0&1\\
1&1&0&&0&1\\
1&1&0&&1&0\\
1&1&1&&0&0\\
0&0&0&0&0&0\\
0&0&0&1&1&0\\
0&0&1&0&1&1\\
0&0&1&1&1&0\\
0&0&1&1&1&1\\
0&1&0&0&0&1\\
0&1&0&1&0&1\\
0&1&0&1&1&1\\
0&1&1&0&0&0\\
0&1&1&0&0&1\\
0&1&1&0&1&1\\
0&1&1&1&0&0\\
0&1&1&1&1&1\\
1&0&0&1&1&0\\
1&0&0&1&1&1\\
1&0&1&0&1&1\\
1&0&1&1&0&1\\
1&0&1&1&1&0\\
1&0&1&1&1&1\\
1&1&0&0&0&1\\
1&1&0&0&1&1\\
1&1&0&1&0&0\\
1&1&0&1&0&1\\
1&1&0&1&1&1\\
1&1&1&0&0&1\\
1&1&1&0&1&0\\
1&1&1&0&1&1
\end{vmatrix}=\begin{vmatrix}
0&0&1&&0&1\\
0&1&0&&1&0\\
0&1&0&&1&1\\
0&1&1&&0&1\\
1&0&0&&1&1\\
1&0&1&&0&1\\
1&1&0&&0&1\\
1&1&0&&1&0\\
1&1&1&&0&0\\
0&0&0&0&0&0\\
0&0&0&1&1&0\\
0&0&1&0&1&1\\
0&0&1&1&1&0\\
0&0&1&1&1&1\\
0&1&0&0&0&1\\
0&1&0&1&0&1\\
0&1&1&0&0&0\\
0&1&1&0&1&1\\
0&1&1&1&0&0\\
0&1&1&1&1&1\\
1&0&0&1&1&0\\
1&0&1&0&1&1\\
1&0&1&1&1&0\\
1&0&1&1&1&1\\
1&1&0&0&1&1\\
1&1&0&1&0&0\\
1&1&0&1&0&1\\
1&1&0&1&1&1\\
1&1&1&0&0&1\\
1&1&1&0&1&0\\
1&1&1&0&1&1\\
1&1&0&1&1&0\\
1&1&1&0&0&0
\end{vmatrix}=\begin{vmatrix}
0&&1&&0&1\\
0&1&0&&1&\\
1&0&0&&1&1\\
1&0&1&&0&1\\
1&1&0&&0&1\\
1&1&0&&1&0\\
1&1&1&&0&0\\
0&0&0&0&0&0\\
0&0&0&1&1&0\\
0&0&1&0&1&1\\
0&0&1&1&1&0\\
0&0&1&1&1&1\\
0&1&0&0&0&1\\
0&1&0&1&0&1\\
0&1&1&0&0&0\\
0&1&1&0&1&1\\
0&1&1&1&0&0\\
0&1&1&1&1&1\\
1&0&0&1&1&0\\
1&0&1&0&1&1\\
1&0&1&1&1&0\\
1&0&1&1&1&1\\
1&1&0&&1&1\\
1&1&0&1&&\\
1&1&1&0&&
\end{vmatrix}=\begin{vmatrix}
0&&1&&0&1\\
0&1&0&&1&\\
1&0&0&&1&1\\
1&0&1&&0&1\\
1&1&0&&0&1\\
1&1&0&&1&\\
1&1&1&&0&0\\
0&0&0&0&0&0\\
&0&0&1&1&0\\
&0&1&0&1&1\\
&0&1&1&1&\\
0&1&0&&0&1\\
0&1&1&&0&0\\
0&1&1&&1&1\\
1&1&0&1&&\\
1&1&1&0&&
\end{vmatrix}=\begin{vmatrix}
0&&1&&0&1\\
&1&0&&1&\\
1&0&0&&1&1\\
1&0&1&&0&1\\
1&1&0&&0&1\\
1&1&1&&0&0\\
0&0&0&0&0&0\\
&0&0&1&1&0\\
&0&1&&1&1\\
&0&1&1&1&\\
0&1&0&&0&1\\
0&1&1&&0&0\\
0&1&1&&1&1\\
1&1&0&1&&\\
1&1&1&0&&
\end{vmatrix}=$$

$$=\begin{vmatrix}
0&&1&&0&1\\
&1&0&&1&\\
1&0&0&&1&1\\
1&0&1&&0&1\\
&1&0&&0&1\\
&1&1&&0&0\\
0&0&0&0&0&0\\
&0&0&1&1&0\\
&0&1&&1&1\\
&0&1&1&1&\\
0&&1&&1&1\\
1&1&0&1&&\\
1&1&1&0&&
\end{vmatrix}=\begin{vmatrix}
0&&1&&&1\\
&1&0&&1&\\
1&0&0&&1&1\\
1&0&1&&0&1\\
&1&0&&0&1\\
&1&1&&0&0\\
0&0&0&0&0&0\\
&0&0&1&1&0\\
&0&1&&1&1\\
&0&1&1&1&0\\
&0&1&1&1&1\\
1&1&0&1&&\\
1&1&1&0&&
\end{vmatrix}=\begin{vmatrix}
0&&1&&&1\\
&1&0&&1&\\
1&0&0&&1&1\\
1&0&1&&0&1\\
&1&0&&0&1\\
&1&1&&0&0\\
0&0&0&0&0&0\\
&0&&1&1&0\\
&0&1&&1&1\\
1&1&0&1&&\\
1&1&1&0&&
\end{vmatrix}=\begin{vmatrix}
0&&1&&&1\\
&1&0&&1&\\
1&0&0&&1&1\\
1&0&1&&0&1\\
&1&0&&0&1\\
&1&1&&0&0\\
0&0&0&0&0&0\\
&0&&1&1&0\\
0&0&1&&1&1\\
1&0&1&&1&1\\
1&1&0&1&&\\
1&1&1&0&&
\end{vmatrix}=\begin{vmatrix}
0&&1&&&1\\
&1&0&&1&\\
1&0&&&1&1\\
1&0&1&&0&1\\
&1&0&&0&1\\
&1&1&&0&0\\
0&0&0&0&0&0\\
&0&&1&1&0\\
1&1&0&1&&\\
1&1&1&0&&
\end{vmatrix}=$$

$$=\begin{vmatrix}
0&&1&&&1\\
&1&0&&1&\\
1&0&&&1&1\\
1&0&1&&&1\\
&1&0&&&1\\
&1&1&&0&0\\
0&0&0&0&0&0\\
&0&&1&1&0\\
1&1&0&1&&\\
1&1&1&0&&
\end{vmatrix}=\begin{vmatrix}
0&&1&&&1\\
&1&0&&1&\\
1&0&&&1&1\\
&0&1&&&1\\
&1&0&&&1\\
&1&1&&0&0\\
0&0&0&0&0&0\\
&0&&1&1&0\\
1&1&0&1&&\\
1&1&1&0&&
\end{vmatrix}.$$

У табл. 4 представлені результати мінімізації функції F (x_1, x_2, x_3, x_4, x_5, x_6) (табл. 3) за допомогою Карти Карно, методом Квайна – Мак-Класкі, методом невизначених коефіцієнтів [25] та аналітичним методом.

Таблиця 3
Таблиця істинності логічної функції F (x_1, x_2, x_3, x_4, x_5, x_6) [25]

x_1	x_2	x_3	x_4	x_5	x_6	x_1	x_2	x_3	x_4	x_5	x_6
0	0	1	–	0	1	0	1	1	0	0	1
0	1	0	–	1	0	0	1	1	0	1	1
0	1	0	–	1	1	0	1	1	1	0	0
0	1	1	–	0	1	0	1	1	1	1	1
1	0	0	–	1	1	1	0	0	1	1	0
1	0	1	–	0	1	1	0	0	1	1	1
1	1	0	–	0	1	1	0	1	0	1	1
1	1	0	–	1	0	1	0	1	1	0	1
1	1	1	–	0	0	1	0	1	1	1	0
0	0	0	0	0	0	1	0	1	1	1	1
0	0	0	1	1	0	1	1	0	0	0	1
0	0	1	0	1	1	1	1	0	0	1	1
0	0	1	1	1	0	1	1	0	1	0	0
0	0	1	1	1	1	1	1	0	1	0	1
0	1	0	0	0	1	1	1	0	1	1	1
0	1	0	1	0	1	1	1	1	0	0	1
0	1	0	1	1	1	1	1	1	0	1	0
0	1	1	0	0	0	1	1	1	0	1	1

Таблиця 4
Результат мінімізації функції F (x_1, x_2, x_3, x_4, x_5, x_6) (табл. 3) [25]

Картою Карно, методом Квайна-Мак-Класкі, методом невизначених коефіцієнтів	Аналітичним методом
$F = \bar{x}_1 x_2 x_6 + x_2 \bar{x}_3 x_5 + x_1 \bar{x}_2 x_5 x_6 + \bar{x}_2 x_3 x_6 + x_2 \bar{x}_3 x_6 + x_2 x_3 \bar{x}_4 \bar{x}_5 + x_2 x_3 \bar{x}_5 \bar{x}_6 + \bar{x}_1 \bar{x}_2 \bar{x}_3 \bar{x}_4 \bar{x}_5 \bar{x}_6 + \bar{x}_2 x_4 x_5 \bar{x}_6 + x_1 x_2 \bar{x}_3 x_4 + x_1 x_2 \bar{x}_4 x_5$	$F = \bar{x}_1 x_3 x_6 + x_2 \bar{x}_3 x_5 + x_1 \bar{x}_2 x_5 x_6 + \bar{x}_2 x_3 x_6 + x_2 \bar{x}_3 x_6 + x_2 x_3 \bar{x}_5 \bar{x}_6 + \bar{x}_1 \bar{x}_2 \bar{x}_3 \bar{x}_4 \bar{x}_5 \bar{x}_6 + \bar{x}_2 x_4 x_5 \bar{x}_6 + x_1 x_2 \bar{x}_3 x_4 + x_1 x_2 x_3 \bar{x}_4$

Споглядаючи табл. 4 бачимо, що результатом мінімізації функції F (x_1, x_2, x_3, x_4, x_5, x_6) (табл. 3) аналітичним методом є мінімальна функція, що містить десять мінтермів. Це на один мінтерм менше ніж у [25].

Приклад 10. Мінімізувати логічну функцію F (x_1, x_2, x_3, x_4, x_5, x_6, x_7) аналітичним методом, яка представлена у канонічній формі [30]:

$$F = \sum \begin{pmatrix} 1,\ 3,\ 5,\ 7,\ 20,\ 21,\ 33,\ 61,\ 63,\ 73,\ 75,\ 77, \\ 79,\ 81,\ 83,\ 85,\ 96,\ 97,\ 99,\ 124,\ 125,\ 127 \end{pmatrix}. \quad (19)$$

Рішення:

$$F(A,B,C,D,E,F,G)=$$

$$=\left|\begin{array}{c|ccccccc}
1 & 0 & 0 & 0 & 0 & 0 & 0 & 1\\
3 & 0 & 0 & 0 & 0 & 0 & 1 & 1\\
5 & 0 & 0 & 0 & 0 & 1 & 0 & 1\\
7 & 0 & 0 & 0 & 0 & 1 & 1 & 1\\
20 & 0 & 0 & 1 & 0 & 1 & 0 & 0\\
21 & 0 & 0 & 1 & 0 & 1 & 0 & 1\\
33 & 0 & 1 & 0 & 0 & 0 & 0 & 1\\
61 & 0 & 1 & 1 & 1 & 1 & 0 & 1\\
63 & 0 & 1 & 1 & 1 & 1 & 1 & 1\\
73 & 1 & 0 & 0 & 1 & 0 & 0 & 1\\
75 & 1 & 0 & 0 & 1 & 0 & 1 & 1\\
77 & 1 & 0 & 0 & 1 & 1 & 0 & 1\\
79 & 1 & 0 & 0 & 1 & 1 & 1 & 1\\
81 & 1 & 0 & 1 & 0 & 0 & 0 & 1\\
83 & 1 & 0 & 1 & 0 & 0 & 1 & 1\\
85 & 1 & 0 & 1 & 0 & 1 & 0 & 1\\
96 & 1 & 1 & 0 & 0 & 0 & 0 & 0\\
97 & 1 & 1 & 0 & 0 & 0 & 0 & 1\\
99 & 1 & 1 & 0 & 0 & 0 & 1 & 1\\
124 & 1 & 1 & 1 & 1 & 1 & 0 & 0\\
125 & 1 & 1 & 1 & 1 & 1 & 0 & 1\\
127 & 1 & 1 & 1 & 1 & 1 & 1 & 1
\end{array}\right|=\left|\begin{array}{ccccccc}
0 & 0 & 0 & 0 & & & 1\\
0 & 0 & 1 & 0 & 1 & 0 & \\
0 & 1 & 0 & 0 & 0 & 0 & 1\\
 & 1 & 1 & 1 & 1 & & 1\\
1 & 0 & 0 & 1 & & & 1\\
1 & 0 & 1 & 0 & 0 & & 1\\
1 & 0 & 1 & 0 & 1 & 0 & 1\\
1 & 1 & 0 & 0 & 0 & 0 & 0\\
1 & 1 & 0 & 0 & 0 & & 1\\
1 & 1 & 1 & 1 & 1 & 0 & 0
\end{array}\right|=$$

$$=\left|\begin{array}{ccccccc}
0 & 0 & 0 & 0 & & & 1\\
0 & 0 & 1 & 0 & 1 & 0 & \\
0 & 1 & 0 & 0 & 0 & 0 & 1\\
 & 1 & 1 & 1 & 1 & & 1\\
1 & 0 & 0 & 1 & & & 1\\
1 & 0 & 1 & 0 & 0 & & 1\\
1 & 0 & 1 & 0 & & 0 & 1\\
1 & 1 & 0 & 0 & 0 & 0 & \\
1 & 1 & 0 & 0 & 0 & & 1\\
1 & 1 & 1 & 1 & 1 & 0 &
\end{array}\right|=\left|\begin{array}{ccccccc}
0 & 0 & 0 & 0 & & & 1\\
0 & 0 & 1 & 0 & 1 & 0 & \\
 & 1 & 0 & 0 & 0 & 0 & 1\\
 & 1 & 1 & 1 & 1 & & 1\\
1 & 0 & 0 & 1 & & & 1\\
1 & 0 & 1 & 0 & 0 & & 1\\
1 & 0 & 1 & 0 & & 0 & 1\\
1 & 1 & 0 & 0 & 0 & 0 & \\
1 & 1 & 0 & 0 & 0 & & 1\\
1 & 1 & 1 & 1 & 1 & 0 &
\end{array}\right|.$$

У табл. 5 представлені результати мінімізації функції F (x_1, x_2, x_3, x_4, x_5, x_6, x_7) (19) за допомогою Карти Карно [30] та аналітичним методом.

Таблиця 5
Результат мінімізації функції F (x_1, x_2, x_3, x_4, x_5, x_6, x_7) (19)

Табличним методом (Карта Карно)	Аналітичним методом
$F = \overline{A}\,\overline{B}\,\overline{C}\,\overline{D}G + \overline{A}\,\overline{B}C\overline{D}\,\overline{E}\,\overline{F} +$ $+ \overline{A}B\overline{C}\,\overline{D}\,\overline{E}\,\overline{F}G + BCDEG +$ $+ A\overline{B}\,\overline{C}DG + A\overline{B}C\overline{D}\,\overline{E}G +$ $+ A\overline{B}C\overline{D}\,\overline{F}G + AB\overline{C}\,\overline{D}\,\overline{E}\,\overline{F} +$ $+ AB\overline{C}\,\overline{D}\,\overline{E}G + ABCDE\overline{F}$	$F = \overline{A}\,\overline{B}\,\overline{C}\,\overline{D}G + \overline{A}\,\overline{B}C\overline{D}E\overline{F} +$ $+ A\overline{C}\,\overline{D}\,\overline{E}\,\overline{F}G + BCDEG +$ $+ A\overline{B}\,\overline{C}DG + A\overline{B}C\overline{D}\,\overline{E}G +$ $+ A\overline{B}C\overline{D}\,\overline{F}G + AB\overline{C}\,\overline{D}\,\overline{E}\,\overline{F} +$ $+ AB\overline{C}\,\overline{D}\,\overline{E}G + ABCDE\overline{F}$

Табл. 5 демонструє, що результатом мінімізації функції F (x_1, x_2, x_3, x_4, x_5, x_6, x_7) (19) аналітичним методом є мінімальна функція, що містить 57 літералів. Це на один літерал менше ніж у [30].

Результати мінімізації функції F (x_1, x_2, x_3, x_4, x_5, x_6, x_7) (19) за допомогою програмного забезпечення Logic Friday 1.1.4 [26] та аналітичного методу співпадають.

Приклад 11. Мінімізувати логічну функцію F (x_1, x_2, x_3, x_4, x_5, x_6, x_7) аналітичним методом, яка задана у канонічній формі [30]:

$$F = \sum \begin{pmatrix} 1,\ 3,\ 5,\ 7,\ 8,\ 9,\ 10,\ 18,\ 20,\ 21,\ 23,\ 24, \\ 29,\ 31,\ 33,\ 35,\ 37,\ 39,\ 40,\ 43,\ 48,\ 51, \\ 53,\ 55,\ 57,\ 59,\ 61,\ 63,\ 64,\ 65,\ 66,\ 69, \\ 73,\ 77,\ 79,\ 81,\ 83,\ 84,\ 85,\ 89,\ 91,\ 93, \\ 97,\ 99,\ 101,\ 102,\ 103,\ 105,\ 106,\ 108, \\ 113,\ 115,\ 116,\ 118,\ 121,\ 123,\ 124,\ 127 \end{pmatrix}. \tag{20}$$

Примітка: значення в Σ є мінтерми для рядків таблиці істинності, коли функція F (x_1, x_2, x_3, x_4, x_5, x_6, x_7) повертає «1» на виході.

Рішення:

Визначимо стек логічних операцій першої матриці функції F (x_1, x_2, x_3, x_4, x_5, x_6, x_7) (20) наступним чином. Побудувавши таблицю істинності функції (20) та провівши обчислення окремо кількості одиниць і нулів у кожному стовпчику побудованої таблиці (згідно з алгоритмом у [12]) можна встановити, що найбільша кількість одиниць знаходиться у крайньому правому стовпчику. Наступним кроком об’єднаємо набори змінних, які містять одиницю у крайній правій позиції набору в окрему матрицю. Така матриця представлена у (21) першою. В іншу окрему матрицю об’єднаємо набори змінних функції (20), які містять нулі у крайній правій позиції набору. Така матриця представлена у (22) першою. Спрощення функції (20) у кожній матриці проводиться окремо.

Матриця зі спільними одиницями у крайньому правому стовпчику та її спрощення має наступний вигляд (21):

(21)

Матриця зі спільними нулями у крайньому правому стовпчику та її спрощення має такий вигляд (22):

$$\left|\begin{array}{c|ccccccc} 8 & 0 & 0 & 0 & 1 & 0 & 0 & 0 \\ 10 & 0 & 0 & 0 & 1 & 0 & 1 & 0 \\ 18 & 0 & 0 & 1 & 0 & 0 & 1 & 0 \\ 20 & 0 & 0 & 1 & 0 & 1 & 0 & 0 \\ 24 & 0 & 0 & 1 & 1 & 0 & 0 & 0 \\ 40 & 0 & 1 & 0 & 1 & 0 & 0 & 0 \\ 48 & 0 & 1 & 1 & 0 & 0 & 0 & 0 \\ 64 & 1 & 0 & 0 & 0 & 0 & 0 & 0 \\ 66 & 1 & 0 & 0 & 0 & 0 & 1 & 0 \\ 84 & 1 & 0 & 1 & 0 & 1 & 0 & 0 \\ 102 & 1 & 1 & 0 & 0 & 1 & 1 & 0 \\ 106 & 1 & 1 & 0 & 1 & 0 & 1 & 0 \\ 108 & 1 & 1 & 0 & 1 & 1 & 0 & 0 \\ 116 & 1 & 1 & 1 & 0 & 1 & 0 & 0 \\ 118 & 1 & 1 & 1 & 0 & 1 & 1 & 0 \\ 124 & 1 & 1 & 1 & 1 & 1 & 0 & 0 \end{array}\right| =$$

$$= \left|\begin{array}{cccccc} 0 & 0 & 0 & 1 & 0 & \\ 0 & 0 & 1 & 0 & 0 & 1 \\ & 0 & 1 & 0 & 1 & 0 \\ 0 & 0 & 1 & 1 & 0 & 0 \\ 0 & 1 & 0 & 1 & 0 & 0 \\ 0 & 1 & 1 & 0 & 0 & 0 \\ 1 & 0 & 0 & 0 & 0 & \\ 1 & 1 & 0 & 0 & 1 & 1 \\ 1 & 1 & 0 & 1 & 0 & 1 \\ 1 & 1 & & 1 & 1 & 0 \\ 1 & 1 & 1 & 0 & 1 & \end{array}\right| = \left|\begin{array}{cccccc} 0 & 0 & 0 & 1 & 0 & \\ 0 & 0 & 1 & 0 & 0 & 1 \\ & 0 & 1 & 0 & 1 & 0 \\ 0 & 0 & & 1 & 0 & 0 \\ 0 & & 0 & 1 & 0 & 0 \\ 0 & 1 & 1 & 0 & 0 & 0 \\ 1 & 0 & 0 & 0 & 0 & \\ 1 & 1 & & 0 & 1 & 1 \\ 1 & 1 & 0 & 1 & 0 & 1 \\ 1 & 1 & & 1 & 1 & 0 \\ 1 & 1 & 1 & 0 & 1 & \end{array}\right|. \qquad (22)$$

Об'єднаємо результати мінімізації двох матриць, враховуючи біти крайнього правого стовпчика перших матриць у (21) та (22), отримаємо МДНФ (23) заданої булевої функції (20).

$$F_{\text{МДНФ}} =$$

$$= \left|\begin{array}{ccccccc}
0 & & 0 & 0 & & & 1 \\
 & 0 & 0 & & 0 & 0 & 1 \\
0 & & 1 & & 1 & & 1 \\
0 & 1 & & & 0 & 1 & 1 \\
0 & 1 & 1 & 1 & & & 1 \\
1 & 0 & & & & 0 & 1 \\
1 & 0 & 0 & 1 & 1 & & 1 \\
1 & & 1 & & 0 & & 1 \\
 & 1 & 0 & 0 & & & 1 \\
1 & & & & 0 & 0 & 1 \\
 & 1 & 1 & 1 & & 1 & 1
\end{array}\right| . \left|\begin{array}{ccccccc}
0 & 0 & 0 & 1 & 0 & & 0 \\
0 & 0 & 1 & 0 & 0 & 1 & 0 \\
 & 0 & 1 & 0 & 1 & 0 & 0 \\
0 & 0 & & 1 & 0 & 0 & 0 \\
0 & & 0 & 1 & 0 & 0 & 0 \\
0 & 1 & 1 & 0 & 0 & 0 & 0 \\
1 & 0 & 0 & 0 & 0 & & 0 \\
1 & 1 & & 0 & 1 & 1 & 0 \\
1 & 1 & 0 & 1 & 0 & 1 & 0 \\
1 & 1 & & 1 & 1 & 0 & 0 \\
1 & 1 & 1 & 0 & 1 & & 0
\end{array}\right| . \qquad (23)$$

Мінімальна функція (23) складається з 22-х мінтермів та містить 118 літералів.

Результат мінімізації функції F $(x_1, x_2, x_3, x_4, x_5, x_6, x_7)$ (20) за допомогою Карти Карно [30] є такий:

$$\begin{aligned}
&F=A'C'D'G+A'B'C'DE'F'+A'B'C'DE'G'+A'B'CD'E'FG'+\\
&+B'CD'EF'+A'B'CEG+A'B'CDE'F'G'+A'BCG+\\
&+A'BC'DE'G'+AB'C'D'E'G'+AB'F'G+AB'C'DEG+\\
&+ACE'F+ABCEF'G'+ABCD'EG'+ABCDFG+BC'D'G+ABC'D'EF+\\
&+ABC'E'F'G+ABC'DEF'G'+ABC'DE'FG'.
\end{aligned} \qquad (24)$$

Однак тестування кодом № 43 – 0101011 функція (24) не дає одиницю. Через що, у виразі мінімальної функції (24) [30] буде допущена помилка.

Результат мінімізації функції F $(x_1, x_2, x_3, x_4, x_5, x_6, x_7)$ (20) за допомогою програмного забезпечення Python [31] є такий:

$$\begin{aligned}
&F=A'B'C'DE'G'+A'B'CD'E'FG'+B'CD'EF'+A'B'DE'F'G'+\\
&+A'C'DE'F'G'+A'BE'FG+A'BCD'E'F'G'+AB'C'D'E'G'+\\
&+AB'C'DEG+ABC'DE'FG'+ABDEF'G'+BCDFG+C'D'F'G+\\
&+ACE'G+A'CEG+A'C'D'G+B'C'E'F'G+BC'D'G+ABD'EFG'+\\
&+A'BCDG+AB'F'G+AE'F'G+ACD'EF'G'.
\end{aligned} \qquad (25)$$

Мінімальна функція (25) вміщує 23 мінтерми, що на один мінтерм більше, порівняно з мінімальною функцією, отриманою аналітичним методом (23).

Результати мінімізації функції F (x_1, x_2, x_3, x_4, x_5, x_6, x_7) (20) за допомогою програмного забезпечення Logic Friday 1.1.4 [26] та аналітичного методу співпадають.

Приклад 12. Мінімізувати 64-розрядну булеву функцію, представлену у ДДНФ, аналітичним методом (табл. 6) [28].

У табл. 6 кольором виділені повні бінарні комбінаторні системи 2-(n, b)-design: 15 систем 2-(2, 4)-design і 1 система 2-(1, 2)-design. Для мінімізації функції потрібно провести 15 логічних операцій супер-склеювання змінних та одну операцію простого склеювання. Після цього потрібно ще здійснити дві послідовні операції напівсклеювання змінних, спочатку у третьому рядку зверху, а потім у четвертому рядку зверху, отримуємо мінімальну функцію (26):

$$
\begin{array}{l}
0000000000011101101001110 01\sim\sim 1010000000010011100011011101 1110101 \\
0000000000100011110101011101 00100\sim\sim 0000100100000001011110011 0011 \\
0000000010011100011011101111010 1000\sim 000110011110011111010000 1100 \\
000000001001110001101110111101010000100110011110011111\sim 100001100 \\
0000000010011100011011101111010 10001\sim 001100111100111110100001100 \\
000000001010001000111100011100100000000110100001001000 0111\sim 1\sim 011 \\
000000001010001\sim\sim 0111100011100100000000110100001001000011 1110011 \\
000000010010000000101111001100 1100\sim\sim 000010100010001111000111 0010 \\
000000010111111100001001101111010\sim\sim 000001111110010011011011 10100 \\
00000001100000001010000101010001 1000000\sim\sim 01000011010100010110 0010 \\
000000011011111100000000001101 11000\sim\sim 00000011111010001001000111101 \\
000000011110111010000100000000110\sim\sim 00000000000111001011010000010 \\
000001100000111\sim 1\sim 0101011001100000000111000011000100111101011001 \\
\sim\sim 00011001100111001000000000000010000001110110010010110010100 00000 \\
0000011001110101 0\sim\sim 0001011001001000000110011101110001000100101000 \\
00\sim\sim 01101101011011101011001100000000001111101010111111100001110001 \\
00\sim\sim 01110111010010001011101010010000011011110110100110001 1101000 \\
0010011000110111111100110111 10000\sim\sim 00011100110100000001101000 01001
\end{array}
\tag{26}
$$

Результати спрощення функції (табл. 6) за допомогою програмного забезпечення, що використовує файли ASCII для імпорту і експорту даних та ґрунтується на методі Квайна – Мак-Класкі [28], і аналітичного методу співпадають за числом мінтермів. Однак у [28] у четвертому рядку зверху мінімальної функції не позначена логічна операція над змінною, тому потрібно вважати, що мінімальна функція (26) містить на один літерал менше.

Таблиця 6

Таблиця істинності 64-розрядної ДДНФ логічної функції [28]

000000000001110110100111001**00**1010000000001001110001101110111110101
000000000001110110100111001**01**1010000000001001110001101110111110101
000000000001110110100111001**10**1010000000001001110001101110111110101
000000000001110110100111001**11**1010000000001001110001101110111110101
0000000000100011110101011101001**00**000010010000000010111100110011
0000000000100011110101011101001**01**000010010000000010111100110011
0000000000100011110101011101001**10**000010010000000010111100110011
0000000000100011110101011101001**11**000010010000000010111100110011
00000000100111000110111011110101000000001100111100111110100001100
00000000100111000110111011110101000010011001111001111111100001100
0000000010011100011011101111010100001**0**001100111100111110100001100
0000000010011100011011101111010100001**1**001100111100111110100001100
0000000010100010001111000111001000000001101000010010000111**0**1**0**011
0000000010100010001111000111001000000001101000010010000111**0**1**1**011
0000000010100010001111000111001000000001101000010010000111**1**1**0**011
000000001010001**00**0111100011100100000000110100001001000011110011
0000000010100010001111000111001000000001101000010010000111**1**1**1**011
000000001010001**01**0111100011100100000000110100001001000011110011
000000001010001**10**0111100011100100000000110100001001000011110011
000000001010001**11**0111100011100100000000110100001001000011110011
00000001001000000001011110011001100**00**0000101000100011110001110010
00000001001000000001011110011001100**01**0000101000100011110001110010
00000001001000000001011110011001100**10**0000101000100011110001110010
00000001001000000001011110011001100**11**0000101000100011110001110010
000000010111111100001001101111010**00**00000111111001001101101110100
000000010111111100001001101111010**01**00000111111001001101101110100
000000010111111100001001101111010**10**00000111111001001101101110100
000000010111111100001001101111010**11**00000111111001001101101110100
0000000110000000101000010101000110000000**00**010000110101000101100010
0000000110000000101000010101000110000000**01**010000110101000101100010
0000000110000000101000010101000110000000**10**010000110101000101100010
0000000110000000101000010101000110000000**11**010000110101000101100010
0000000110111111000000000110111000**00**00000011111010001001000111101
0000000110111111000000000110111000**01**00000011111010001001000111101
0000000110111111000000000110111000**10**00000011111010001001000111101
0000000110111111000000000110111000**11**00000011111010001001000111101
000000011110111010000100000000110**00**00000000000111001011010000010
000000011110111010000100000000110**01**00000000000111001011010000010
000000011110111010000100000000110**10**00000000000111001011010000010
000000011110111010000100000000110**11**00000000000111001011010000010
000001100000111**0**1**0**0101011001100000000111000011000100111101011001
000001100000111**0**1**1**0101011001100000000111000011000100111101011001
000001100000111**1**1**0**0101011001100000000111000011000100111101011001
000001100000111**1**1**1**0101011001100000000111000011000100111101011001
0000011001100111001000000000000100000111011001001011001010000000
00000110011101010**00**000101100100100000110011101110001000100101000
00000110011101010**01**000101100100100000110011101110001000100101000
00000110011101010**10**000101100100100000110011101110001000100101000
00000110011101010**11**000101100100100000110011101110001000100101000

Продовження Таблиці 6

00**00**011011010110111010110011000000000111101010111111 00001110001
00**00**011101110100100010111010100100000110111101101001100011101000
00**01**011011010110111010110011000000000111101010111111 00001110001
00**01**011101110100100010111010100100000110111101101001100011101000
0010011000110111111100110111110000**00**000111001101000000110100001001
0010011000110111111100110111110000**01**000111001101000000110100001001
0010011000110111111100110111110000**10**000111001101000000110100001001
0010011000110111111100110111110000**11**000111001101000000110100001001
00**10**011011010110111010110011000000000111101010111111 00001110001
00**10**011101110100100010111010100100000110111101101001100011101000
00**11**011011010110111010110011000000000111101010111111 00001110001
00**11**011101110100100010111010100100000110111101101001100011101000
010001100110011100100000000000000100000111011001001011001010000000
100001100110011100100000000000000100000111011001001011001010000000
110001100110011100100000000000000100000111011001001011001010000000

8. Обговорення результатів мінімізації булевих функцій візуально-матричною формою аналітичного методу

Математичний апарат візуально-матричної форми аналітичного методу розглянуто у роботах [8–10, 12, 32, 33] та ін. Складові методу представлені у табл. 7.

Таблиця 7
Технологія аналітичного методу для візуально-матричної форми

1	Бінарні комбінаторні системи з повторенням 2-(*n*, *b*)-design, 2-(*n*, *x*/*b*)-design
2	Вербальне і образне представлення інформації
3	Логічна операція супер-склеювання змінних
4	Логічна операція неповного супер-склеювання змінних
5	Герменевтика логічних операцій на бінарних структурах
6	Протоколи образних перетворень
7	Ознака мінімальної логічної функції,
8	Мінімізація булевих функцій на повній таблиці істинності
9	Алгоритм аналітичного методу та його автоматизація
10	Поширення аналітичного методу на інші логічні базиси

У даній роботі додані нові складові технології мінімізації булевих функцій візуально-матричною формою аналітичного методу (табл. 8):

Таблиця 8
Нові складові технології мінімізації візуально-матричною формою аналітичного методу

1	Відносно складні алгоритми застосування логічних операцій поглинання та супер-склеювання змінних
2	Стек логічних операцій

При спрощенні логічних функцій не завжди очевидно, який із законів алгебри логіки необхідно застосувати на тому чи іншому кроці. Наглядні структури бінарних матриць та уніфікація оригінальних процедур до певної міри дозволяють вирішувати цю проблему.

Особливістю візуально-матричної форми аналітичного методу є те, що метод ґрунтується на бінарних комбінаторних системах з повторенням 2-(n, b)-design, 2-(n, x/b)-design. Таблиця істинності логічних функцій також являє собою комбінаторну систему з повторенням. Це дозволяє при мінімізації функцій обійтись без допоміжних об'єктів, як то карти Карно, карти Махоні, діаграми Вейча, ациклічний граф, ненаправлений граф, таблиці покриття, куби та ін. Наглядність 2-вимірних бінарних матриць дозволяє здійснювати ручний спосіб спрощення булевих функцій (з використанням математичного редактора, наприклад MathType 7.4.0) у межах до 64 вхідних змінних (приклад 12) для ДДНФ (ДКНФ) представлення функції.

У табл. 9 представлені результати мінімізації булевих функцій, запозичених з робіт інших авторів та аналітичного методу.

Табл. 9 демонструє, що результати мінімізації аналітичного методу, програмного забезпечення JQM – Java Quine McChuskey 1.2.4 [29] та в окремих випадках Logic Friday 1.1.4 [26], співпадають. Для решти прикладів мінімізація аналітичним метод демонструє мінімальну логічну функцію, або меншою на один мінтерм, або меншою на один літерал, або функцію, що вміщує менше на одну інвертовану змінну.

Обмеженням застосування візуально-матричної форми аналітичного методу є випадки, коли перемикальна функція представлена у змішаному базисі. У цьому випадку функцію необхідно представити одним логічним базисом.

Слабка сторона розглянутого методу полягає у малому практичному застосуванні візуально-матричної форми аналітичного методу для мінімізації булевих функцій з подальшим проектуванням і виготовленням відповідних обчислювальних компонентів. Негативні внутрішні фактори методу пов'язані з додатковими часовими витратами на встановлення протоколів спрощення логічних функцій з подальшим створенням бібліотеки протоколів, що мають ілюстрацію відповідних образних перетворень.

Перспективою подальших досліджень може бути пошук нових правил перетворення логічних функцій на основі відстані Гаммінга.

Таблиця 9

Порівняльна таблиця прикладів мінімізації булевих функцій, запозичених з робіт інших авторів та візуально-матричної форми аналітичного методу

№ прикладу	Назва методу мінімізації	Число вхідних змінних	Результат мінімізації	Результат аналітичного методу
7	Карта Карно, методом Квайна-Мак-Класкі, методом невизначених коефіцієнтів [25]	5	6 мінтермів	5 мінтермів
7	Програмне забезпечення Logic Friday 1.1.4 [26]	5	11 інвертованих змінних	10 інвертованих змінних
8	Програмне забезпечення, що використовує файли ASCII для імпорту і експорту даних та ґрунтується на методі Квайна-Мак-Класкі [28]	6	24 інвертованих змінних	23 інвертованих змінних
8	Програмне забезпечення JQM – Java Quine McChuskey 1.2.4 [29]	6	12 мінтермів	12 мінтермів
8	Програмне забезпечення Logic Friday 1.1.4 [26]	6	24 інвертованих змінних	23 інвертованих змінних
9	Карта Карно, методом Квайна-Мак-Класкі, методом невизначених коефіцієнтів [25]	6	11 мінтермів	10 мінтермів
10	Карта Карно [30]	7	58 літералів	57 літералів
10	Програмне забезпечення Logic Friday 1.1.4 [26]	7	Результати мінімізації співпадають	
11	Карта Карно [30]	7	21 мінтерм (не проходить верифікацію)	22 мінтерми
11	Програмне забезпечення Python [31]	7	23 мінтерми	22 мінтерми
11	Програмне забезпечення Logic Friday 1.1.4 [26]	7	Результати мінімізації співпадають	
12	Програмне забезпечення, що використовує файли ASCII для імпорту і експорту даних та ґрунтується на методі Квайна-Мак-Класкі [28]	64	1120 літералів	1119 літералів

9. Висновки

1. Порівняно складніші алгоритми застосування логічних операцій поглинання та супер-склеювання змінних розширюють варіанти їхнього використання, що дозволяє збільшувати ефективність процедури мінімізації булевих функцій.

2. Оптимальне рішення для мінімізації булевих функцій аналітичним методом ґрунтується на першочерговому застосуванні операції супер-склеювання змінних (якщо така операція можлива) у межах таблиці істинності заданої булевої функції [12]. Однак існує невелике число винятків, коли мінімізація є більш ефективна, якщо спочатку застосувати логічну операцію простого склеювання змінних [12]. Таким чином необхідний короткий аналіз для першочергового застосування логічних операцій у першій бінарній матриці. Такий аналіз здійснюється шляхом вибору, за встановленими критеріями, оптимального стеку логічних операцій для першої бінарній матриці, а, в окремих випадках, і для другої бінарної матриці (приклад 5). Це забезпечує належну ефективність мінімізації і до раніш не врахованих варіантів спрощення булевих функцій аналітичним методом.

3. Після спрощення у першій бінарній матриці ДДНФ (ДКНФ), задана функція переходить у ДНФ (КНФ) представлення, стає функцією загального вигляду. Класичним алгоритмом мінімізації булевих функцій загального виду є метод Блейка-Порецького. Однак у переважній більшості випадків достатньо буде провести 1–3 логічних операцій узагальнене склеювання змінних. Мінімізація логічних функцій загального виду можлива і з операції поглинання змінних. Візуально-матрична форма налаштовує до напівінтуїтивного підходу мінімізації булевих функцій, що ґрунтується на деяких властивостей сприйняття бінарних матриць. Саме такі методи потрібні для ручної мінімізації булевих функцій [3].

4. Ефективність візуально-матричної форми аналітичного методу для мінімізації булевих функцій демонструється наступними прикладами:

– *приклад* 3 [13] – мінімізація 4-розрядної булевої функції;
– *приклад* 4 [21] – мінімізація 3-розрядної булевої функції;
– *приклад* 5 [12], *приклад* 7 [25] – мінімізація 5-розрядних булевих функцій;
– *приклад* 8 [28], *приклад* 9 [25] – мінімізація 6-розрядних булевих функцій;
– *приклад* 10 [30], *приклад* 11 [30] – мінімізація 7-розрядних булевих функції.
– *приклад* 12 [28] – мінімізація 64-розрядної булевої функції.

За результатами порівняння встановлено, що ефективність візуально-матричної форми аналітичного методу для мінімізації булевих функцій дає підставу для доцільності застосування її у процедурах мінімізації логічних функцій, оскільки візуально-матрична форма аналітичного методу спроможна:

– забезпечити оперативний вибір стеку логічних операцій у першій та другій бінарних матрицях, що у підсумку дає оптимальний сценарій мінімізації логічних функцій;

– збільшити ефективність процедури мінімізації логічних функцій за рахунок реалізації відносно складних алгоритмів застосування логічних операцій поглинання та супер-склеювання змінних.

Література

1. Налимов, В. В. (1993). В поисках иных смыслов. М.: Издательская группа «Прогрес», 280. URL: https://platona.net/load/knigi_po_filosofii/filosofija_poznanija/nalimov_v_poiskakh_inykh_smyslov/45-1-0-566
2. Глушков, В. М. (1986). Кибернетика. Вопросы теории и практики. М.: Наука, 488. URL: http://www.pseudology.org/science/Glushkov_Kibernetika._Voprosue_teorii_i_practiki.pdf
3. Закревский, А. Д. (1960). Визуально-матричный метод минимизации булевых функций. Автоматика и телемеханика, 21 (3), 369–373. URL: http://www.mathnet.ru/links/fcef8cd452ff8b3804279fce2157d772/at12511.pdf
4. Плехль, О.; Юрасов, А. Н. (Ред.) (1959). Электромеханическая коммутация и коммутационные аппараты. (Контактные схемы и аппараты). Введение в теорию и расчет. М.; Л.: Госэнергоиздат, 288. URL: https://catalogue.nure.ua/document=101624
5. Svoboda, A. (1956). Graficko-mechanické pomůcky užívané při analyse a synthese kontaktových obvodů. Stroje na zpracování informací. Prague: Czechoslovak Academy of Sciences, Research Institute of Mathematical Machines. 9–22.
6. Karnaugh, M. (1953). The map method for synthesis of combinational logic circuits. Transactions of the American Institute of Electrical Engineers, 72, 593–598.
7. Донец, С. (2015). Истоки скрытой информативности образа. Філологічні науки, 21, 107–112. URL: http://dspace.pnpu.edu.ua/handle/123456789/6095
8. Riznyk, V., Solomko, M. (2017). Minimization of Boolean functions by combinatorial method. Technology Audit and Production Reserves, 4 (2 (36)), 49–64. doi: https://doi.org/10.15587/2312-8372.2017.108532
9. Riznyk, V., Solomko, M. (2017). Application of super-sticking algebraic operation of variables for Boolean functions minimization by combinatorial method. Technology Audit and Production Reserves, 6 (2 (38)), 60–76. doi: https://doi.org/10.15587/2312-8372.2017.118336
10. Riznyk, V., Solomko, M. (2018). Research of 5-bit boolean functions minimization protocols by combinatorial method. Technology Audit and Production Reserves, 4 (2 (42)), 41–52. doi: https://doi.org/10.15587/2312-8372.2018.140351
11. Кондратенко, Н. Р. (2010). Комп'ютерний практикум з математичної логіки. Вінниця: ВНТУ, 117. URL: https://www.twirpx.com/file/993689/
12. Riznyk, V., Solomko, M., Tadeyev, P., Nazaruk, V., Zubyk, L., Voloshyn, V. (2020). The algorithm for minimizing Boolean functions using a method of the optimal combination of the sequence of figurative transformations. Eastern-European Journal of Enterprise Technologies, 3 (4 (105)), 43–60. doi: https://doi.org/10.15587/1729-4061.2020.206308
13. Başçiftçi, F., Akar, H. (2020). Smart minterm ordering and accumulation approach for insignificant function minimization. Ain Shams Engineering Journal. doi: https://doi.org/10.1016/j.asej.2020.04.003

14. Bernasconi, A., Ciriani, V., Villa, T. (2020). Exploiting Symmetrization and D-reducibility for Approximate Logic Synthesis. IEEE Transactions on Computers, 1–1. doi: https://doi.org/10.1109/tc.2020.3043476

15. Young, M. H., Muroga, S. (1985). Minimal covering problem and PLA minimization. International Journal of Computer & Information Sciences, 14 (6), 337–364. doi: https://doi.org/10.1007/bf00991179

16. Huang, J. (2014). Programing implementation of the Quine-McCluskey method for minimization of Boolean expression. arXiv.org. URL: https://arxiv.org/ftp/arxiv/papers/1410/1410.1059.pdf

17. El-Bakry, H. M., Mastorakis, N. (2009). A fast computerized method for automatic simplification of boolean functions. Proceedings of the 9th WSEAS International Conference on SYSTEMS THEORY AND SCIENTIFIC COMPUTATION (ISTASC '09), 99–107. URL: https://www.researchgate.net/profile/Hazem_El-Bakry/publication/228877182_A_fast_computerized_method_for_automatic_simplification_of_boolean_functions/links/553fa8230cf29680de9bf997/A-fast-computerized-method-for-automatic-simplification-of-boolean-functions.pdf

18. Duşa, A., Thiem, A. (2015). Enhancing the Minimization of Boolean and Multivalue Output Functions WitheQMC. The Journal of Mathematical Sociology, 39 (2), 92–108. doi: https://doi.org/10.1080/0022250x.2014.897949

19. Dusa, A. (2019). Consistency Cubes: a fast, efficient method for exact Boolean minimization. The R Journal, 10 (2), 357. doi: https://doi.org/10.32614/rj-2018-080

20. Rudell, R. L. (1989). Logic synthesis for VLSI design. Electronics Research Laboratory. URL: http://www.cs.columbia.edu/~cs6861/handouts/rudell-PhD-thesis.pdf

21. Сенчуков, В. Ф., Денисова, Т. В. (2020). v-мінімізація булевих функцій за матрицею відстаней та зведенням до задачі математичного програмування. Відкриті інформаційні та комп'ютерні інтегровані технології, 88, 123–133. doi: https://doi.org/10.32620/oikit.2020.88.10

22. Рицар, Б. Є. (2013). Мінімізація системи логікових функцій методом паралельного розчеплення кон’юнктермів. Вісник Національного університету "Львівська політехніка". Радіоелектроніка та телекомунікації, 766, 18–27. URL: http://nbuv.gov.ua/UJRN/VNULPPT_2013_766_6

23. Поспелов, Д. А. (1974). Логические методы анализа и синтеза схем. М.: «Энергия», 368. URL: http://urss.ru/cgi-bin/db.pl?lang=Ru&blang=ru&page=Book&id=25326

24. Шестаков, В. И. (Ред.) (1954). Синтез электронных вычислительных и управляющих схем. М., 358.

25. Методы минимизации функций алгебры логики. Материал из Национальной библиотеки им. Н. Э. Баумана. URL: https://ru.bmstu.wiki/Методы_минимизации_функций_алгебры_логики

26. Logic Friday 1.1.4. URL: https://www.softpedia.com/get/Others/Home-Education/Logic-Friday.shtml

27. Минимизатор эвристической логики эспрессо. URL: https://ru.qaz.wiki/wiki/Espresso_heuristic_logic_minimizer

28. Boolean functions' minimisation software based on the Quine-McCluskey method. URL: http://www.seattlerobotics.org/encoder/200106/qmccmin.htm

29. JQM Java Quine McCluskey. URL: https://sourceforge.net/projects/jqm-java-quine-mccluskey/

30. Kumar, V. D. A., Amuthan, S. G. (2016). Static structure simplification of boolean function for “n” variables - a novel approach. ICTACT Journal on Microelectronics, 1 (4), 160–167. doi: https://doi.org/10.21917/ijme.2016.0024

31. Web service Python. URL: https://trinket.io/python/fbbf7518b8

32. Riznyk, V., Solomko, M. (2018). Minimization of conjunctive normal forms of boolean functions by combinatorial method. Technology Audit and Production Reserves, 5 (2 (43)), 42–55. doi: https://doi.org/10.15587/2312-8372.2018.146312

33. Solomko, M., Khomiuk, N., Ivashchuk, Y., Nazaruk, V., Reinska, V., Zubyk, L., Popova, A. (2020). Implementation of the method of image transformations for minimizing the Sheffer functions. Eastern-European Journal of Enterprise Technologies, 5 (4 (107)), 19–34. doi: https://doi.org/10.15587/1729-4061.2020.214899