

УДК 519.171:510.28

Н. А. Якімова, М. Є. Клішин

Одеський національний університет ім. І.І. Мечникова

МАТРИЧНЕ ПОДАННЯ ОПЕРАЦІЙ НАД ГРАФАМИ

Теорія графів має широке розповсюдження з практичної точки зору. Графи оточують нас у повсякденному житті (наприклад, карти доріг та шляхів), а також відіграють важливу роль у наукових дослідженнях (наприклад, електросхеми). Для побутового застосування, безумовно, найзручнішим є геометричний спосіб подання графів. Але для комп'ютерної обробки інформації це не є раціональним. В цих випадках використовується матричне подання графів у вигляді матриць суміжності або матриць інцидентності. Тому все більшого значення набувають дослідження, присвячені саме цій темі. У статті розглядається можливість виконання операцій над матрицями, якими подано графи. Ці методи мають свої особливості та обмеження. Вони також розглянуті у статті. Для кожної операції запропонований варіант обробки як матриці суміжності, так і матриці інцидентності для орієнтованих та неорієнтованих графів, показано відмінності такої обробки в залежності від виду графу.

MSC: 03G05, 03G25, 03F52, 06E25, 15B34.

Ключові слова: орієнтований та неорієнтований граф, матриця суміжності, матриця інцидентності, операції над графами, елементарні логічні операції, булева матриця, багатозначна логіка.

DOI: 10.18524/2519-206X.2022.1-2(39-40).294314.

Вступ

Сьогодні ставить перед наукою все більшу кількість задач, пов'язаних із комп'ютеризацією. Саме потреби комп'ютерної обробки інформації потребують дослідження можливостей матричного подання графів і, як наслідок, матричного подання усіх можливих перетворень графів. Операції над графами – це перший важливий крок на шляху таких перетворень. Матричне подання цих операцій має свої особливості в залежності від виду кожного конкретного графу, тому процедура їх виконання не є повністю універсальною. Різні види графів потребують своїх індивідуальних підходів до їх розглядання. За матрицями суміжності та інцидентності можна повністю охарактеризувати граф та відновити його геометричну реалізацію. Для простих неорієнтованих графів ці матриці є булевими. Якщо граф не має кратних ребер, то його матриця суміжності є булевою незалежно

від того, орієнтований цей граф чи ні. Матриця інцидентності є булевою лише для неорієнтованого графу. Тому для матричного виконання операцій над графами треба залучати не лише апарат булевих матриць [1], а й апарат багатозначної логіки та апарат звичайних арифметичних операцій.

ПОПЕРЕДНІ РЕЗУЛЬТАТИ. Однією з важливіших алгебраїчних моделей є апарат матриць, який може бути застосований в багатьох як теоретичних, так і прикладних галузях математики.

Означення. Матриця називається булевою, якщо її елементами є логічні скаляри із поля $K=\{0,1\}$ [1].

Тобто елементами булевої матриці є нулі та одиниці. Наприклад, булевою буде матриця [2]

$$A = \begin{pmatrix} 1 & 0 & 1 \\ 1 & 1 & 0 \\ 0 & 0 & 1 \end{pmatrix}.$$

Якщо матриці є булевими, то з ними можна виконувати як звичайні алгебраїчні операції над матрицями, так і операції двозначної логіки, описані в [1]. Якщо матриці не є булевими, то для виконання з ними логічних операцій диз'юнкції та кон'юнкції треба застосовувати апарат багатозначної логіки. В цьому випадку операції диз'юнкції та кон'юнкції матриць виконуються за наступними правилами [3]

$$x \vee y = \max\{x, y\}, \quad (1)$$

$$x \wedge y = \min\{x, y\}. \quad (2)$$

ОСНОВНІ РЕЗУЛЬТАТИ

Можливість матричного виконання основних операцій над графами можна показати на прикладах. Розглянемо два орієнтовані графи, зображені на рис. 1.

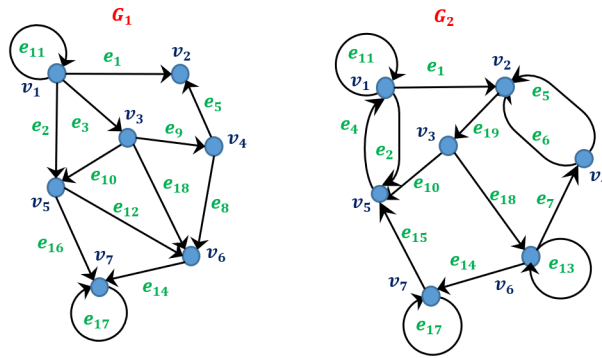


Рис. 1. Орієнтовані графи

Для обох цих графів можна побудувати матриці суміжності:

$$A(G_1) = \begin{pmatrix} 1 & 1 & 1 & 0 & 1 & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 1 & 1 & 1 & 0 \\ 0 & 1 & 0 & 0 & 0 & 1 & 0 \\ 0 & 0 & 0 & 0 & 0 & 1 & 1 \\ 0 & 0 & 0 & 0 & 0 & 0 & 1 \\ 0 & 0 & 0 & 0 & 0 & 0 & 1 \end{pmatrix}, \quad A(G_2) = \begin{pmatrix} 1 & 1 & 0 & 0 & 1 & 0 & 0 \\ 0 & 0 & 1 & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & 1 & 1 & 0 \\ 0 & 2 & 0 & 0 & 0 & 0 & 0 \\ 1 & 0 & 0 & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 1 & 0 & 1 & 1 \\ 0 & 0 & 0 & 0 & 1 & 0 & 1 \end{pmatrix}.$$

Ці матриці побудовано виходячи з того, що номер рядка відповідає початковій вершині, а номер стовпця – кінцевій вершині кожного ребра. Але ці графи можуть бути від самого початку задані цими матрицями. Виконання операцій над такими графами не потребує відтворення їх геометричної реалізації.

Граф G_1 не містить кратних ребер, тому його матриця суміжності є булевою. Граф G_2 містить суворо паралельні ребра $e_5(v_4, v_2)$ і $e_6(v_4, v_2)$. Тому його матриця суміжності містить елемент $a_{42} = 2$, тобто булевою не є. Але якщо врахувати, що операція диз'юнкції для багатозначної логіки виконується за правилом (1), то диз'юнкція для цих матриць набуває вигляду:

$$\begin{aligned}
 A(G_1) \vee A(G_2) &= \begin{pmatrix} 1 & 1 & 1 & 0 & 1 & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 1 & 1 & 1 & 0 \\ 0 & 1 & 0 & 0 & 0 & 1 & 0 \\ 0 & 0 & 0 & 0 & 0 & 1 & 1 \\ 0 & 0 & 0 & 0 & 0 & 0 & 1 \\ 0 & 0 & 0 & 0 & 0 & 0 & 1 \end{pmatrix} \vee \begin{pmatrix} 1 & 1 & 0 & 0 & 1 & 0 & 0 \\ 0 & 0 & 1 & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & 1 & 1 & 0 \\ 0 & 2 & 0 & 0 & 0 & 0 & 0 \\ 1 & 0 & 0 & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 1 & 0 & 1 & 1 \\ 0 & 0 & 0 & 0 & 1 & 0 & 1 \end{pmatrix} = \\
 &= \begin{pmatrix} 1 & 1 & 1 & 0 & 1 & 0 & 0 \\ 0 & 0 & 1 & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 1 & 1 & 1 & 0 \\ 0 & 2 & 0 & 0 & 0 & 1 & 0 \\ 1 & 0 & 0 & 0 & 0 & 1 & 1 \\ 0 & 0 & 0 & 1 & 0 & 1 & 1 \\ 0 & 0 & 0 & 0 & 1 & 0 & 1 \end{pmatrix} = A_U.
 \end{aligned}$$

Якщо операцію об'єднання вказаних графів виконати графічно, то отримаємо граф, зображений на рис. 2.

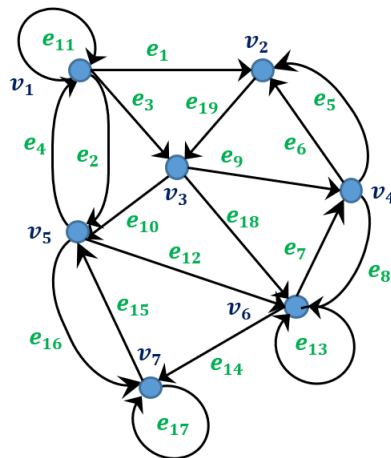


Рис. 2. Граф $G_1 \cup G_2$

Легко побачити, що цьому графу відповідає матриця суміжності A_U .

Якщо врахувати, що для багатозначної логіки операція кон'юнкції виконується за правилом (2), то кон'юнкція для матриць $A(G_1)$ і $A(G_2)$ набуває вигляду:

$$\begin{aligned}
 A(G_1) \wedge A(G_2) &= \begin{pmatrix} 1 & 1 & 1 & 0 & 1 & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 1 & 1 & 1 & 0 \\ 0 & 1 & 0 & 0 & 0 & 1 & 0 \\ 0 & 0 & 0 & 0 & 0 & 1 & 1 \\ 0 & 0 & 0 & 0 & 0 & 0 & 1 \\ 0 & 0 & 0 & 0 & 0 & 0 & 1 \end{pmatrix} \wedge \begin{pmatrix} 1 & 1 & 0 & 0 & 1 & 0 & 0 \\ 0 & 0 & 1 & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & 1 & 1 & 0 \\ 0 & 2 & 0 & 0 & 0 & 0 & 0 \\ 1 & 0 & 0 & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 1 & 0 & 1 & 1 \\ 0 & 0 & 0 & 0 & 1 & 0 & 1 \end{pmatrix} = \\
 &= \begin{pmatrix} 1 & 1 & 0 & 0 & 1 & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & 1 & 1 & 0 \\ 0 & 1 & 0 & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 & 0 & 1 \\ 0 & 0 & 0 & 0 & 0 & 0 & 1 \end{pmatrix} = A_{\cap}.
 \end{aligned}$$

Якщо операцію перетину вказаних графів виконати графічно, то отримаємо граф, зображений на рис. 3.

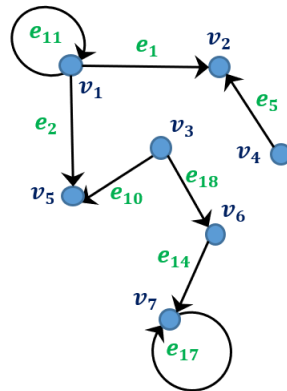


Рис. 3. Граф $G_1 \cap G_2$

Легко побачити, що цьому графу відповідає матриця суміжності A_{\cap} .

Виконаємо тепер з матрицями $A(G_1)$ і $A(G_2)$ операцію суми за модулем 2, маючи на увазі її загальне означення як залишку від ділення відповідної суми на 2:

$$\begin{aligned}
 A(G_1) \oplus A(G_2) &= \begin{pmatrix} 1 & 1 & 1 & 0 & 1 & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 1 & 1 & 1 & 0 \\ 0 & 1 & 0 & 0 & 0 & 1 & 0 \\ 0 & 0 & 0 & 0 & 0 & 1 & 1 \\ 0 & 0 & 0 & 0 & 0 & 0 & 1 \\ 0 & 0 & 0 & 0 & 0 & 0 & 1 \end{pmatrix} \oplus \begin{pmatrix} 1 & 1 & 0 & 0 & 1 & 0 & 0 \\ 0 & 0 & 1 & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & 1 & 1 & 0 \\ 0 & 2 & 0 & 0 & 0 & 0 & 0 \\ 1 & 0 & 0 & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 1 & 0 & 1 & 1 \\ 0 & 0 & 0 & 0 & 1 & 0 & 1 \end{pmatrix} = \\
 &= \begin{pmatrix} 0 & 0 & 1 & 0 & 0 & 0 & 0 \\ 0 & 0 & 1 & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 1 & 0 & 0 & 0 \\ 0 & 1 & 0 & 0 & 0 & 1 & 0 \\ 1 & 0 & 0 & 0 & 0 & 1 & 1 \\ 0 & 0 & 0 & 1 & 0 & 1 & 0 \\ 0 & 0 & 0 & 0 & 1 & 0 & 0 \end{pmatrix} = A_{\oplus}.
 \end{aligned}$$

Якщо виконати графічно операцію кільцевої суми графів G_1 і G_2 , то отримаємо граф, зображений на рис. 4.

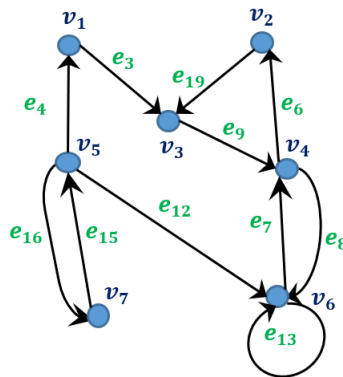


Рис. 4. Граф $G_1 \oplus G_2$

Для цього графу також легко побачити, що йому відповідає матриця суміжності A_{\oplus} .

В цих графах кожному ребру наданий порядковий номер. В практичному застосуванні ці номери можуть означати якесь певне змістовне навантаження. Але при матричному відображенні цей зміст може втрачатися. Матриця відображає наявність або відсутність ребра, тобто наявність або відсутність зв'язку між об'єктами. Тому, якщо в двох графах між двома вершинами одне й те саме ребро має різне змістовне навантаження (наприклад, автомобільна та ґрунтова дороги), то матриця суміжності покаже лише наявність або відсутність цього зв'язку без пояснення його характеру. Але зазвичай у практичних застосунках інформації про наявність зв'язку буває достатньо, тому для бінарних операцій над орієнтованими графами [4] застосування елементарних операцій багатозначної логіки над матрицями суміжності є ефективним математичним апаратом.

Застосуємо тепер означення кільцевої суми графів. Згідно з ним, множина ребер кільцевої суми графів визначається зі співвідношення $E_{\oplus} = \{(E_{G_1} \cup E_{G_2}) \setminus (E_{G_1} \cap E_{G_2})\}$ [4]. Виходячи з цього співвідношення, виконаємо арифметичну операцію віднімання матриці A_{\wedge} від матриці A_{\vee} :

$$\begin{aligned}
 A_{\cup} - A_{\cap} &= \begin{pmatrix} 1 & 1 & 1 & 0 & 1 & 0 & 0 \\ 0 & 0 & 1 & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 1 & 1 & 1 & 0 \\ 0 & 2 & 0 & 0 & 0 & 1 & 0 \\ 1 & 0 & 0 & 0 & 0 & 1 & 1 \\ 0 & 0 & 0 & 1 & 0 & 1 & 1 \\ 0 & 0 & 0 & 0 & 1 & 0 & 1 \end{pmatrix} - \begin{pmatrix} 1 & 1 & 0 & 0 & 1 & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & 1 & 1 & 0 \\ 0 & 1 & 0 & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 & 0 & 1 \\ 0 & 0 & 0 & 0 & 0 & 0 & 1 \end{pmatrix} = \\
 &= \begin{pmatrix} 0 & 0 & 1 & 0 & 0 & 0 & 0 \\ 0 & 0 & 1 & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 1 & 0 & 0 & 0 \\ 0 & 1 & 0 & 0 & 0 & 1 & 0 \\ 1 & 0 & 0 & 0 & 0 & 1 & 1 \\ 0 & 0 & 0 & 1 & 0 & 1 & 0 \\ 0 & 0 & 0 & 0 & 1 & 0 & 0 \end{pmatrix}.
 \end{aligned}$$

Ця матриця збігається з матрицею A_{\oplus} для кільцевої суми графів G_1 і G_2 , тобто матриця суміжності графа, отриманого в результаті кільцевої суми двох графів, може бути обчисленою і в такий спосіб. Цей приклад наочно показує, що для матричного виконання операцій над графами одночасне застосування і арифметичних, і логічних операцій є припустимим. Для програмної реалізації перетворень матриць це також не є перешкодою. Однак слід зазначити, що як арифметичні, так і логічні операції над матрицями висувають певні вимоги до розмірності цих матриць: операції арифметичного додавання і віднімання, а також логічні операції диз'юнкції, кон'юнкції та суми за модулем 2 можна виконувати лише з матрицями однакової розмірності [1]. З цього випливає, що в описаний спосіб можна виконувати операції в матричному поданні лише для графів, у яких збігаються множини вершин, причому не лише за кількістю, а й за змістом.

Слід також зазначити, що кільцева сума графів не може містити ізольованих вершин. Якщо при виконанні цієї операції такі вершини утворюються, то їх треба видаляти з графа.

Видалення вершини з графу тягне за собою видалення всіх інцидентних до неї ребер, тобто видалення всіх зв'язків цього об'єкта або вузла з іншими об'єктами або вузлами. Це означає, що при видаленні вершини v_i з матриці суміжності треба видалити i -й рядок та i -й стовпець. У зв'язку з цим алгоритм виконання операції видалення вершини v_i з графа в матричному поданні аналогічний алгоритму побудови мінору M_{ii} для матриці суміжності цього графу. Нехай, наприклад, з графу G_1 треба видалити вершину v_4 . Побудуємо для матриці $A(G_1)$ мінор M_{44} [5]:

$$M_{44} = \begin{vmatrix} 1 & 1 & 1 & 1 & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 1 & 1 & 0 \\ 0 & 0 & 0 & 0 & 1 & 1 \\ 0 & 0 & 0 & 0 & 0 & 1 \\ 0 & 0 & 0 & 0 & 0 & 1 \end{vmatrix}.$$

Отже, новому графу $G'_1 = G_1 \setminus \{v_4\}$ буде відповідати матриця суміжності

$$A(G'_1) = \begin{pmatrix} 1 & 1 & 1 & 1 & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 1 & 1 & 0 \\ 0 & 0 & 0 & 0 & 1 & 1 \\ 0 & 0 & 0 & 0 & 0 & 1 \\ 0 & 0 & 0 & 0 & 0 & 1 \end{pmatrix}.$$

Алгоритм видалення з матриці рядків та стовпців вже комп'ютеризований. У даному випадку програмна реалізація буде передбачати виконання двох зсувів: для рядків та стовпців. Для графа без ізольованих вершин доцільно проводити цю операцію саме з використанням матриці суміжності.

У матриці суміжності ознакою ізольованої вершини є наявність одиниць у рядку і стовпці. У матриці інцидентності ознакою ізольованої вершини є наявність нульового рядка. Отже, якщо вершина, яку треба видалити з графу, є ізольованою, то в цьому єдиному випадку зручніше проводити цю операцію з використанням матриці інцидентності. При видаленні нульового рядка з матриці інцидентності жодний зв'язок між іншими вершинами (об'єктами або вузлами) не порушується. Тому не потрібно відслідковувати появу стовпців, які після видалення цього рядка будуть містити лише одну одиницю, що не є припустимим для матриці інцидентності [4]. Використання матриці інцидентності при видаленні ізольованої вершини скорочує вдвічі програмну реалізацію (не треба одночасно видалити ще й стовпець, до того ж зсув буде теж лише один). Нехай, наприклад, граф G_3 заданий своєю матрицею суміжності $A(G_3)$:

$$A(G_3) = \begin{pmatrix} 0 & 1 & 1 & 0 & 1 \\ 1 & 0 & 1 & 0 & 1 \\ 1 & 1 & 0 & 0 & 1 \\ 0 & 0 & 0 & 0 & 0 \\ 1 & 1 & 1 & 0 & 0 \end{pmatrix}.$$

Матриця $A(G_3)$ містить повністю нульові 4-й рядок та 4-й стовпець. Це свідчить про ізольованість вершини v_4 . Ця матриця повністю симетрична, тому граф G_3 є неорієнтованим. Його можна вважати повністю заданим верхньотрикутною частиною матриці $A(G_3)$. У відповідності з

алгоритмом, описаним в [4], за цією матрицею можна побудувати матрицю інцидентності $I(G_3)$ графа G_3 без відновлення його геометричної реалізації. Таким чином, неорієнтованому графу G_3 відповідає матриця інцидентності

$$I(G_3) = \begin{pmatrix} 1 & 1 & 1 & 0 & 0 & 0 \\ 1 & 0 & 0 & 1 & 1 & 0 \\ 0 & 1 & 0 & 1 & 0 & 1 \\ 0 & 0 & 0 & 0 & 0 & 0 \\ 0 & 0 & 1 & 0 & 1 & 1 \end{pmatrix}.$$

В матриці інцидентності $I(G_3)$ ізольованій вершині v_4 відповідає нульовий 4-й рядок. Щоб видалити з графа G_3 цю ізольовану вершину треба із матриці $I(G_3)$ видалити лише 4-й рядок без подальшого додаткового видалення низки стовпців. Таким чином, для графа $G'_3 = G_3 \setminus \{v_4\}$ маємо матрицю інцидентності

$$I(G'_3) = \begin{pmatrix} 1 & 1 & 1 & 0 & 0 & 0 \\ 1 & 0 & 0 & 1 & 1 & 0 \\ 0 & 1 & 0 & 1 & 0 & 1 \\ 0 & 0 & 1 & 0 & 1 & 1 \end{pmatrix}.$$

За потреби за цією матрицею можна відновити геометричну реалізацію графа G'_3 або його матрицю суміжності [4].

Операція видалення одного ребра передбачає зникнення із графу якогось одного ребра (зв'язку або відношення) зі збереженням всіх вершин (об'єктів або вузлів) та решти зв'язків між об'єктами [6]. Таким чином, у матриці суміжності змін зазнають лише ті елементи, які відповідають ребрам, що видаляються. Решта елементів матриці суміжності зберігають свої значення. Порядок матриці також залишається тим самим, бо ніяка вершина внаслідок видалення ребра із графа не зникає. При видаленні орієнтованого ребра зменшується лише один елемент матриці суміжності. При видаленні неорієнтованого ребра однакових змін зазнає пара симетричних елементів матриці суміжності. Нехай між вершинами v_i і v_j необхідно видалити певну кількість l_{ij}^- ребер. Отже, для виконання цієї операції треба виконати операцію арифметичного віднімання матриці A_e^- від матриці суміжності початкового графу. У матриці-від'ємнику A_e^- елементи,

що відповідають ребрам, які видаляються, будуть дорівнювати l_{ij}^- . Решта елементів цієї матриці будуть нульовими.

Розглянемо, наприклад, граф $G_1 \cup G_2$, якому відповідає матриця суміжності A_{\cup} . В цій матриці є пара симетричних елементів $a_{46}=a_{64}=1$. Ці елементи відповідають парі протилежно спрямованих ребер, яку можна замінити одним неорієнтованим ребром. Нехай з цього графу треба видалити це неорієнтоване ребро. Нехай до того ж з цього графа треба видалити також одне з суворо паралельних орієнтованих ребер від вершини v_4 до вершини v_2 . Це означає, що в матриці A_e^- відмінними від нуля будуть лише елементи $l_{42}^- = 1$ і $l_{46} = l_{64} = 1$. Отже, щоб обчислити матрицю суміжності графа

$$(G_1 \cup G_2)' = (G_1 \cup G_2) \setminus \{ \vec{e}(v_4, v_2) \} \setminus \{ e(v_4, v_6) \}$$

(запис \vec{e} означає, що дане ребро є орієнтованим), можна отримати, виконавши наступне арифметичне віднімання:

$$\begin{aligned} A_{\cup} - A_e^- &= \begin{pmatrix} 1 & 1 & 1 & 0 & 1 & 0 & 0 \\ 0 & 0 & 1 & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 1 & 1 & 1 & 0 \\ 0 & 2 & 0 & 0 & 0 & 1 & 0 \\ 1 & 0 & 0 & 0 & 0 & 1 & 1 \\ 0 & 0 & 0 & 1 & 0 & 1 & 1 \\ 0 & 0 & 0 & 0 & 1 & 0 & 1 \end{pmatrix} - \begin{pmatrix} 0 & 0 & 0 & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 & 0 & 0 \\ 0 & 1 & 0 & 0 & 0 & 1 & 0 \\ 0 & 0 & 0 & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 1 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 & 0 & 0 \end{pmatrix} = \\ &= \begin{pmatrix} 1 & 1 & 1 & 0 & 1 & 0 & 0 \\ 0 & 0 & 1 & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 1 & 1 & 1 & 0 \\ 0 & 1 & 0 & 0 & 0 & 0 & 0 \\ 1 & 0 & 0 & 0 & 0 & 1 & 1 \\ 0 & 0 & 0 & 0 & 0 & 1 & 1 \\ 0 & 0 & 0 & 0 & 1 & 0 & 1 \end{pmatrix} = A((G_1 \cup G_2)'). \end{aligned}$$

Програмна реалізація цієї операції значно простіша, ніж її алгебраїчне обґрунтування. Для її виконання достатньо зменшити на величини l_{ij}^- відповідні елементи матриці суміжності початкового графа.

Операція доповнення визначається для неорієнтованих графів. Тому, якщо граф є орієнтованим або змішаним, спочатку треба побудувати для нього асоційований (або співвіднесений) граф [4], а потім вже для цього нового графу виконувати операцію доповнення. З матриці суміжності $A(G)$ початкового графа обчислюємо нову матрицю $A^s(G)$ за правилом:

$$a_{ij}^s = a_{ji}^s = \min \{1, \max \{a_{ij}, a_{ji}\}\}. \quad (3)$$

Отримана матриця буде булевою. Але вона може містити одиниці на головній діагоналі, тобто відображати граф з петлями. Співвіднесений граф, який не може містити петлі, використовується в цих обчисленнях лише як проміжний результат. Тому позбутися цих петель можна один раз вже на останньому кроці, а не щоразу для всіх проміжних матриць. Наступним кроком для отримання матриці суміжності доповнення графа G треба виконати логічну операцію заперечення [1] матриці $A^s(G)$, тобто обчислити матрицю $\overline{A^s(G)}$.

Для прикладу знайдемо матрицю суміжності доповнення орієнтованого графа G_2 . Побудуємо для цього графа матрицю $A^s(G_2)$:

$$A^s(G_2) = \begin{pmatrix} 1 & 1 & 0 & 0 & 1 & 0 & 0 \\ 1 & 0 & 1 & 1 & 0 & 0 & 0 \\ 0 & 1 & 0 & 0 & 1 & 1 & 0 \\ 0 & 1 & 0 & 0 & 0 & 1 & 0 \\ 1 & 0 & 1 & 0 & 0 & 0 & 1 \\ 0 & 0 & 1 & 1 & 0 & 1 & 1 \\ 0 & 0 & 0 & 0 & 1 & 1 & 1 \end{pmatrix}.$$

Тепер обчислимо заперечення цієї матриці, тобто матрицю $\overline{A^s(G)}$:

$$\overline{A^s(G_2)} = \begin{pmatrix} \bar{1} & \bar{1} & \bar{0} & \bar{0} & \bar{1} & \bar{0} & \bar{0} \\ \bar{1} & \bar{0} & \bar{1} & \bar{1} & \bar{0} & \bar{0} & \bar{0} \\ \bar{0} & \bar{1} & \bar{0} & \bar{0} & \bar{1} & \bar{1} & \bar{0} \\ \bar{0} & \bar{1} & \bar{0} & \bar{0} & \bar{0} & \bar{1} & \bar{0} \\ \bar{1} & \bar{0} & \bar{1} & \bar{0} & \bar{0} & \bar{0} & \bar{1} \\ \bar{0} & \bar{0} & \bar{1} & \bar{1} & \bar{0} & \bar{1} & \bar{1} \\ \bar{0} & \bar{0} & \bar{0} & \bar{0} & \bar{1} & \bar{1} & \bar{1} \end{pmatrix} = \begin{pmatrix} 0 & 0 & 1 & 1 & 0 & 1 & 1 \\ 0 & 1 & 0 & 0 & 1 & 1 & 1 \\ 1 & 0 & 1 & 1 & 0 & 0 & 1 \\ 1 & 0 & 1 & 1 & 1 & 0 & 1 \\ 0 & 1 & 0 & 1 & 1 & 1 & 0 \\ 1 & 1 & 0 & 0 & 1 & 0 & 0 \\ 1 & 1 & 1 & 1 & 0 & 0 & 0 \end{pmatrix}.$$

Тепер треба позбутися петель в отриманому графі, тобто одиниць на головній діагоналі отриманої матриці. Для цього треба виконати її кон'юнкцію із запереченням одиничної матриці. В результаті буде отримано матрицю суміжності графа-доповнення:

$$\begin{aligned} \overline{A^s(G_2)} \wedge \overline{E} &= \begin{pmatrix} 0 & 0 & 1 & 1 & 0 & 1 & 1 \\ 0 & 1 & 0 & 0 & 1 & 1 & 1 \\ 1 & 0 & 1 & 1 & 0 & 0 & 1 \\ 1 & 0 & 1 & 1 & 1 & 0 & 1 \\ 0 & 1 & 0 & 1 & 1 & 1 & 0 \\ 1 & 1 & 0 & 0 & 1 & 0 & 0 \\ 1 & 1 & 1 & 1 & 0 & 0 & 0 \end{pmatrix} \wedge \begin{pmatrix} 0 & 1 & 1 & 1 & 1 & 1 & 1 \\ 1 & 0 & 1 & 1 & 1 & 1 & 1 \\ 1 & 1 & 0 & 1 & 1 & 1 & 1 \\ 1 & 1 & 1 & 0 & 1 & 1 & 1 \\ 1 & 1 & 1 & 1 & 0 & 1 & 1 \\ 1 & 1 & 1 & 1 & 1 & 0 & 1 \\ 1 & 1 & 1 & 1 & 1 & 1 & 0 \end{pmatrix} = \\ &= \begin{pmatrix} 0 & 0 & 1 & 1 & 0 & 1 & 1 \\ 0 & 0 & 0 & 0 & 1 & 1 & 1 \\ 1 & 0 & 0 & 1 & 0 & 0 & 1 \\ 1 & 0 & 1 & 0 & 1 & 0 & 1 \\ 0 & 1 & 0 & 1 & 0 & 1 & 0 \\ 1 & 1 & 0 & 0 & 1 & 0 & 0 \\ 1 & 1 & 1 & 1 & 0 & 0 & 0 \end{pmatrix} = A(\overline{G_2}). \end{aligned}$$

Для неорієнтованих графів формула (3) набуває вигляду

$$a_{ij}^s = a_{ji}^s = \min \{1, a_{ij}\}. \quad (4)$$

Для неорієнтованих графів без кратних ребер матрицю суміжності графа-доповнення можна отримати відразу кон'юнкцією заперечення матриці суміжності початкового графа з матрицею \overline{E} .

В булевій алгебрі кон'юнкція з одиницею не впливає на результат. Тому в програмній реалізації на останньому кроці побудови матриці $A(\overline{G})$ можна просто присвоїти нульове значення усім елементам головної діагоналі, тобто всім елементам, для яких збігаються номер стовпця і номер рядка ($i = j$).

Операція введення ребра є зворотною до операції видалення ребра. При введенні орієнтованого ребра передбачається збільшення одного елемента

матриці суміжності. При введенні неорієнтованого ребра однакових змін зазнає пара симетричних елементів матриці суміжності. Нехай між вершинами v_i і v_j необхідно додати певну кількість l_{ij}^+ ребер. При програмній реалізації відбувається збільшення відповідних елементів матриці суміжності початкового графа на вказану величину. Алгебраїчна реалізація цієї операції передбачає арифметичну суму двох матриць. Одним з доданків є матриця суміжності початкового графа. У другому доданку A_e^+ елементи, що відповідають новим ребрам, дорівнюють l_{ij}^+ . Решта елементів другого доданку є нульовими.

Нехай, наприклад, до графу G_1 необхідно ввести два суворо паралельних орієнтованих ребра від вершини v_2 до вершини v_4 і одне неорієнтоване ребро між вершинами v_3 і v_7 . Це означає, що в даному випадку $l_{24}^+ = 2$ і $l_{37}^+ = l_{73}^+ = 1$. Тоді матрицю суміжності графа $G_1'' = G_1 + 2 \cdot \{\vec{e}(v_2, v_4)\} + \{e(v_3, v_7)\}$ можна отримати, виконавши наступну арифметичну суму:

$$\begin{aligned}
 A(G_1) + A_e^+ &= \begin{pmatrix} 1 & 1 & 1 & 0 & 1 & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 1 & 1 & 1 & 0 \\ 0 & 1 & 0 & 0 & 0 & 1 & 0 \\ 0 & 0 & 0 & 0 & 0 & 1 & 1 \\ 0 & 0 & 0 & 0 & 0 & 0 & 1 \\ 0 & 0 & 0 & 0 & 0 & 0 & 1 \end{pmatrix} + \begin{pmatrix} 0 & 0 & 0 & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 2 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 & 0 & 1 \\ 0 & 0 & 0 & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 & 0 & 0 \\ 0 & 0 & 1 & 0 & 0 & 0 & 0 \end{pmatrix} = \\
 &= \begin{pmatrix} 1 & 1 & 1 & 0 & 1 & 0 & 0 \\ 0 & 0 & 0 & 2 & 0 & 0 & 0 \\ 0 & 0 & 0 & 1 & 1 & 1 & 1 \\ 0 & 1 & 0 & 0 & 0 & 1 & 0 \\ 0 & 0 & 0 & 0 & 0 & 1 & 1 \\ 0 & 0 & 0 & 0 & 0 & 0 & 1 \\ 0 & 0 & 1 & 0 & 0 & 0 & 1 \end{pmatrix} = A(G_1'').
 \end{aligned}$$

Операцію введення вершини в ребро розглянемо також на прикладі графа G_1 . Її зручніше виконувати з матрицею інцидентності. Цю матрицю для графа G_1 можна отримати із його матриці суміжності або безпосередньо із

геометричної реалізації. Для зручності в матриці інцидентності було збережено ту нумерацію ребер, яку подано для цього графа на рис. 1. Для наочності запишемо цю матрицю у вигляді таблиці.

$$I(G_1) =$$

	e_1	e_2	e_3	e_5	e_8	e_9	e_{10}	e_{11}	e_{12}	e_{14}	e_{16}	e_{17}	e_{18}
v_1	1	1	1	0	0	0	0	±1	0	0	0	0	0
v_2	-1	0	0	-1	0	0	0	0	0	0	0	0	0
v_3	0	0	-1	0	0	1	1	0	0	0	0	0	1
v_4	0	0	0	1	1	-1	0	0	0	0	0	0	0
v_5	0	-1	0	0	0	0	-1	0	1	0	1	0	0
v_6	0	0	0	0	-1	0	0	0	-1	1	0	0	-1
v_7	0	0	0	0	0	0	0	0	0	-1	-1	±1	0

Нехай треба ввести вершину w' в ребро $e_2(v_1, v_5)$ і вершину w'' в ребро $e_{14}(v_6, v_7)$. Обидва ці ребра є орієнтованими. При введенні вершини в ребро напрямком руху новими ребрами має збігатися з напрямком, який був у старого ребра. Схему алгоритму перетворення матриці інцидентності для виконання цієї операції подано на рис. 5.

	e_1	e_2	e'_2	e''_2	e_3	e_5	e_8	e_9	e_{10}	e_{11}	e_{12}	e_{14}	e'_{14}	e''_{14}	e_{16}	e_{17}	e_{18}
v_1	1	1	1	0	1	0	0	0	0	±1	0	0	0	0	0	0	0
w'	0	0	-1	1	0	0	0	0	0	0	0	0	0	0	0	0	0
v_2	-1	0	0	0	-1	0	0	0	0	0	0	0	0	0	0	0	0
v_3	0	0	0	0	-1	0	0	1	1	0	0	0	0	0	0	0	1
v_4	0	0	0	0	0	1	1	-1	0	0	0	0	0	0	0	0	0
v_5	0	-1	0	-1	0	0	0	0	-1	0	1	0	0	0	1	0	0
v_6	0	0	0	0	0	0	-1	0	0	0	-1	1	1	0	0	0	-1
w''	0	0	0	0	0	0	0	0	0	0	0	0	-1	1	0	0	0
v_7	0	0	0	0	0	0	0	0	0	0	0	-1	0	-1	-1	±1	0

Рис. 5. Схема перетворення матриці інцидентності при введенні вершини в ребро

При введенні вершини w' ребро $e_2(v_1, v_5)$ з графа видаляється. Аналогічно при введенні вершини w'' видаляється ребро $e_{14}(v_6, v_7)$. На схемі відповідні цим ребрам стовпці матриці інцидентності зафарбовано чорним кольором. Замість ребра $e_2(v_1, v_5)$ з'являються 2 нові ребра (два нові стовпця e'_2 і e''_2 в матриці). Рух цими ребрами має повторювати напрямок руху ребром e_2 . Тому початкова вершина ребра e'_2 має збігатися з початковою

вершиною ребра e_2 . З кінцевою вершиною ребра e_2 має збігатися кінцева вершина ребра e_2'' . Таким чином, нова вершина w' є транзитною при русі від вершини v_1 до вершини v_5 . Тобто вона є кінцевою для нового ребра e_2' і початковою для нового ребра e_2'' . Рядок для нової вершини в матриці має з'явитися в проміжку між рядками, що відповідають граничним вершинам старого ребра. Зручніше додавати його відразу після початкової вершини, тобто в даному випадку після вершини v_1 . При цьому елементи « -1 » в стовпці e_2' і елемент « 1 » в стовпці e_2'' будуть єдиними елементами в рядку цієї вершини.

Аналогічно замість ребра $e_{14}(v_6, v_7)$ також з'являються два нові ребра (нові стовпці e_{14}' і e_{14}'' в матриці). Вершина v_6 є початковою для старого ребра e_{14} і нового ребра e_{14}' . Вершина v_7 є кінцевою для старого ребра e_{14} і нового ребра e_{14}'' . Рядок для нової транзитної вершини w'' буде містити лише два ненульові елементи: « -1 » у стовпці e_{14}' і « 1 » у стовпці e_{14}'' . Цей новий рядок буде розташований між рядками v_6 і v_7 . Рядки нових вершин на схемі рис. 5 зафарбовано зеленим кольором, а стовпці нових ребер – сірим. Таким чином, матриця інцидентності нового графа буде мати вигляд:

$$I(G_1''') =$$

	e_1	e_2'	e_2''	e_3	e_5	e_8	e_9	e_{10}	e_{11}	e_{12}	e_{14}'	e_{14}''	e_{16}	e_{17}	e_{18}
v_1	1	1	0	1	0	0	0	0	±1	0	0	0	0	0	0
w'	0	-1	1	0	0	0	0	0	0	0	0	0	0	0	0
v_2	-1	0	0	0	-1	0	0	0	0	0	0	0	0	0	0
v_3	0	0	0	-1	0	0	1	1	0	0	0	0	0	0	1
v_4	0	0	0	0	1	1	-1	0	0	0	0	0	0	0	0
v_5	0	0	-1	0	0	0	0	-1	0	1	0	0	1	0	0
v_6	0	0	0	0	0	-1	0	0	0	-1	1	0	0	0	-1
w''	0	0	0	0	0	0	0	0	0	0	-1	1	0	0	0
v_7	0	0	0	0	0	0	0	0	0	0	0	-1	-1	±1	0

За потреби від цієї матриці можна перейти до матриці суміжності $A(G_1''')$ або геометричної реалізації графа G_1''' .

У випадку, коли нова вершина вводиться до неорієнтованого ребра, алгоритм перетворення матриці інцидентності буде таким самим. Відмінність буде полягати лише в тому, що нові ребра будуть також неорієнтованими. Тому всі ненульові елементи нових стовпців будуть додатними, тобто дорівнювати « $+1$ ».

Використаємо матрицю інцидентності $I(G_1)$, щоб дослідити алгоритм

її перетворення при виконанні операції замикання (ототожнення) вершин. Ця операція можлива лише для суміжних вершин. У графі G_1 такими вершинами, наприклад, є вершини v_1 і v_5 (їх зв'язує ребро e_2), а також вершини v_2 і v_4 (їх зв'язує ребро e_5). Ототожнимо кожну з цих пар вершин. Згідно з означенням цієї операції [4], усі вершини, суміжні хоча б з однією з ототожнюваних вершин, будуть суміжні з новою вершиною. Це означає, що усі ребра, інцидентні хоча б з однією з ототожнюваних вершин, зберігаються в графі і будуть інцидентні новій вершині. Ребро, що зв'язує ототожнювані вершини, перетворюється на петлю. Таким чином, у матриці $I(G_1)$ має з'явитися новий рядок, що відповідає новій вершині. Його елементи є арифметичною сумою відповідних елементів рядків тих вершин, для яких проводиться операція замикання. Рядки пари ототожнюваних вершин при цьому з матриці видаляються. Схему цих перетворень показано на рис. 6.

	e_1	e_2	e_3	e_5	e_8	e_9	e_{10}	e_{11}	e_{12}	e_{14}	e_{16}	e_{17}	e_{18}	
$v_{1,5}$	1	±1	1	0	0	0	-1	±1	1	0	1	0	0	$v_1 + v_5$
$v_{2,4}$	-1	0	0	±1	1	-1	0	0	0	0	0	0	0	$v_2 + v_4$
v_1	1	1	1	0	0	0	0	±1	0	0	0	0	0	
v_2	-1	0	0	-1	0	0	0	0	0	0	0	0	0	
v_3	0	0	-1	0	0	1	1	0	0	0	0	0	1	
v_4	0	0	0	1	1	-1	0	0	0	0	0	0	0	
v_5	0	-1	0	0	0	0	-1	0	1	0	1	0	0	
v_6	0	0	0	0	-1	0	0	0	-1	1	0	0	-1	
v_7	0	0	0	0	0	0	0	0	0	-1	-1	±1	0	

Рис. 6. Схема перетворення матриці інцидентності при замиканні вершин

Ребра e_2 і e_5 перетворилися на петлі (на схемі відповідні стовпці зафарбовано рожевим). Рядки, що відповідають вершинам, які було замкнено (v_1 з v_5 і v_2 з v_4), з матриці інцидентності видаляються (на схемі ці рядки зафарбовано чорним). У матриці з'являються нові рядки, що відповідають новим вершинам – результатам замикання. На схемі ці рядки зафарбовано зеленим. Елементи рядка $v_{1,5}$ дорівнюють арифметичній сумі відповідних елементів рядків v_1 і v_5 , а елементи рядка $v_{2,4}$ дорівнюють арифметичній сумі відповідних елементів рядків v_2 і v_4 . Таким чином, матриця інцидентності нового графа $G_1^{v^+}$ має вигляд:

$$I(G_1^{v^+}) =$$

	e_1	e_2	e_3	e_5	e_8	e_9	e_{10}	e_{11}	e_{12}	e_{14}	e_{16}	e_{17}	e_{18}
$v_{1,5}$	1	± 1	1	0	0	0	-1	± 1	1	0	1	0	0
$v_{2,4}$	-1	0	0	± 1	1	-1	0	0	0	0	0	0	0
v_3	0	0	-1	0	0	1	1	0	0	0	0	0	1
v_6	0	0	0	0	-1	0	0	0	-1	1	0	0	-1
v_7	0	0	0	0	0	0	0	0	0	-1	-1	± 1	0

У разі, якщо граф є неорієнтованим, то схема перетворення його матриці інцидентності залишається такою ж самою, але замість арифметичної суми елементів ототожнених вершин виконується операція суми за модулем 2 цих елементів. Як і при виконанні попередніх операцій, за потреби від отриманої матриці інцидентності можна перейти до матриці суміжності досліджуваного графа або відновити його геометричну реалізацію.

Операція стягування ребра відбувається за тим самим алгоритмом, що й операція замикання вершин, але з подальшим видаленням з графа отриманих петель. Таким чином, при стягуванні ребра спочатку відбувається ототожнення вершин, що є граничними для цього ребра. Отже, якщо для графа G_1 стоїть задача стягнути ребра e_2 і e_5 , то можна також скористатися схемою на рис. 6. Спочатку за цією схемою замикаються пари вершин v_1 і v_5 (є граничними для ребра e_2), а також v_2 і v_4 (є граничними для ребра e_5). В результаті цього замикання утворюються дві петлі (рожеві стовпці на схемі), які треба видалити з графа. Як наслідок, із матриці інцидентності треба видалити стовпці e_2 і e_5 . Петлі, що не мають відношення до жодної з ототожнених вершин, у графі залишаються. Тому в матриці ці стовпці також залишаються без змін (стовпці e_{11} і e_{17}). В результаті буде отримана матриця інцидентності нового графа $G_1^{e^+}$:

$$I(G_1^{e^+}) =$$

	e_1	e_3	e_8	e_9	e_{10}	e_{11}	e_{12}	e_{14}	e_{16}	e_{17}	e_{18}
$v_{1,5}$	1	1	0	0	-1	± 1	1	0	1	0	0
$v_{2,4}$	-1	0	1	-1	0	0	0	0	0	0	0
v_3	0	-1	0	1	1	0	0	0	0	0	1
v_6	0	0	-1	0	0	0	-1	1	0	0	-1
v_7	0	0	0	0	0	0	0	-1	-1	± 1	0

Висновки

Деякі операції зручніше виконувати з матрицями суміжності, а деякі — з матрицями інцидентності. Одні й ті самі алгоритми мають відмінності в залежності від того, орієнтовані чи вільні графи беруть участь у розглянутих операціях. В залежності від видів графів також є обмеження на відображення змістовної інформації матрицями цих графів. Але в практичних застосунках ці обмеження, як правило, несуттєві. Отже, для кожної операції над графами та кожного виду графів можна запропонувати комбінацію алгебраїчних операцій (арифметичних та логічних), що дозволяють отримати матрицю нового графа, або чіткий легко програмований алгоритм перетворення матриць початкових графів. Жодна з розглянутих операцій над графами не є неможливою в матричному виконанні. Запропоновані алгоритми можуть значно спростити комп'ютерну обробку графів.

Список літератури

1. **Гвоздинская Н. А.** О логических матрицах / Н. А. Гвоздинская, З. В. Дударь, С. А. Пославский, Ю. П. Шабанов-Кушнарченко // Проблемы бионики. — 1998. — Т.48. — С. 12–22.
2. **Якімова Н. А.** Предикатні логічні матриці / Н. А. Якімова // Вісник Одеського національного університету ім.І. І. Мечнікова. Дослідження в математиці і механіці. — 2019. — Т.24, №2 (34). — С. 67–74.
3. **Шаповров С. Д.** Математическая логика. Курс лекций и практических занятий / С. Д. Шаповров. — СПб.: БХВ-Петербург, 2005. — 416 с.
4. **Якімова Н. А.** Дискретна математика. Частина 1. Теорія множин. Теорія графів. Курс лекцій / Н. А. Якімова. — Одеса: ОНУ ім. І. І. Мечникова, 2022. — 102 с.
5. **Зиков О. О.** Лекції з алгебри / О. О. Зиков. — Одеса: Астропринт, 2007. — 400 с.
6. **Капітонова Ю. В.** Основи дискретної математики / Ю. В. Капітонова, С. Л. Кривий, О. А. Летичевський, та ін. — Київ: Наукова думка, 2002. — 580 с.

Yakimova N.A., Klishyn M.E.

MATRIX REPRESENTATION OF OPERATIONS ON GRAPHS

Summary

This article considers the possibility of matrix execution of both unary and binary operations on graphs. Graphs, as an abstract mathematical construction, have a very wide range of practical applications. First of all, it is algorithmization and computer processing of information, electrical engineering, etc. Therefore, it is important to have a mathematical apparatus that allows you to transform the graphical presentation of information about objects into algebraic models for their further research using purely mathematical methods. If necessary, it is always possible to return from such an algebraic model to a graphical representation of the object (for example, to a graphical representation of a circuit diagram in electronics). For each of the considered operations on graphs, either a combination of algebraic operations or an easily programmable matrix processing algorithm is proposed, which can be used to represent any graph. Attention is also paid to the differences in such processing depending on the type of graphs involved in the considered operations. Some operations are more convenient to perform with adjacency matrices, and some - with incidence matrices. This article also considers these features of matrix execution of operations on graphs. All the proposed algorithms are illustrated with specific detailed examples. Thus, it is shown that for all operations on graphically presented objects, their matrix interpretation is possible. This result greatly facilitates the possibility of software implementation of work with such graphic objects.

Key words: directed and undirected graph, adjacency matrix, incidence matrix, operations on graphs, elementary logical operations, Boolean matrix, multivalued logic.

REFERENCES

1. Gvozdinskaya, N. A., Dudar, Z. V., Poplavskiy, S. A. and Shabanov-Kushnarenko, Y. P. (1998). O logicheskikh matrityakh [On logical matrix]. *Problemy bioniki*, Vol.48, P. 12–22.
2. Yakimova, N. A. (2019). Predykatni logichni matritzi [Predicative logical matrices]. *Visnyk Odes'kogo Natsional'nogo universitetu. Researches in Mathematics and Mechanics*, Vol.24, №2(34), P. 67–74.

3. Shaporev, S. D. (2005). *Matematicheskaya logika [Mathematical logic]*. St-Peterburg: BHV-Peterburg, 416 p.
4. Yakimova, N. A. (2021). *Dyskretna matematyka. Chastyna 1. Teoriya mnozhin. Teoriya grafiv [Discrete Math. Part 1. Set theory. Graf theory]*. Odessa: ONU im.I.I.Mechnikova, 102 p.
5. Zykov, A. A. (2007). *Lektzii z algebry [Lectures on algebra]*. Odessa: Astroprint, 400 p.
6. Kapitonova, Yu. V., Kryvyy, S. L., Letychevskiy, O. A., Lutzkiy, G. M. and Pechurin, M. K. (2002). *Osnovy dyskretnoi matematyky [Foundation of Discrete Mathematics]*. Kyiv: Naukova dumka, 578 p.