

ВІДТВОРЕННЯ ГРАФІВ ЗА ТЕХНОЛОГІЧНИМИ ШЛЯХАМИ

За допомогою утворюючих систем графів розв'язується проблема відтворення графів за їх підграфами. Запропоновано алгоритм відтворення графів. Розглянуто приклад.

С помощью образующих систем графов решена проблема восстановления графов по их подграфам. Предложен алгоритм восстановления графов. Рассмотрен пример.

It is solved the problem of graphs regenerating by their subgraphs with the help of deriving systems of graphs. In the article there is offered the algorithm of solving this problem and considered an example of graphs regenerating.

Вступ

Існує досить широкий клас різноманітних задач проектування, планування, штучного інтелекту тощо, які зводяться до розгляду графових моделей. Але в цьому класі існує підклас задач, для яких графові моделі вдається визначити лише частково або не повністю з причин неповноти інформації і інших. Наприклад, існує деякий технологічний процес, який реалізується на складній системі (залізничних станцій, автостанцій України тощо), для якого загальна технологічна модель не відома, але відомі локальні технологічні процеси (між суміжними станціями). Зрозуміло, що в такій ситуації, без відомої загальної моделі, ефективно планування і раціональне виконання технологічного процесу для всієї системи може бути неможливим, або досить складно реалізованим. Тому природно поставити питання: «Як відтворити модель технологічного процесу системи за відомими частковими моделями на окремих елементах цієї системи?». Поставлене питання характеризує проблему відтворення моделі за її підмоделями.

Інші проблеми відтворення моделей виникають в задачах розпізнання образів [1], формальних граматики [2], системах формальних структур [3].

В матеріалах статті розглянуто частковий випадок проблеми відтворення моделей. Зокрема, розглянуто графове представлення технологічних процесів і введено поняття графових структур. З'ясовано, що графові структури не однозначно відтворюють графові моделі. Виходячи з цього поставлена задача відтворення графової моделі технологічного процесу за заданими графовими структурами деякого технологічного зразка. Задача відтворення розв'язується за допомогою введених утворюючих систем підграфів. В загальному вигляді задачу відтворення розглянуто у роботі [4].

Представлення графів

Об'єктом досліджень буде орієнтований граф G , визначений на множині вершин $C = \{c_i; i \in I\}$ з навантаженими дугами [5]. Нехай елементи навантаження дуг графу задаються символами деякого скінченного алфавіту $E = \{e_j; j \in J\}$ з порожнім символом ε . Графи будемо представляти у формульному вигляді так: якщо для дуги (c_i, c_j) графу G навантажену символом $e_k \in E$ ввести позначення $e_{k,i,j}$, то будь який шлях P_i графу можна записати як

$$P_i \triangleq (e_{s,q,k}, e_{j,k,g}, \dots, e_{r,n,m}) \quad (1)$$

У записі (1) елемент $e_{s,q,k}$ назвемо початковим з початком у вершині c_q шляху P_i і відповідно елемент $e_{r,n,m}$ – заключним з заключною вершиною c_m . За записом довільного шляху графу G через зв'язану по другому і третьому індексам послідовність елементів $e_{r,h,v}$ у вигляді (1) завжди можливо однозначно виконати класичне [5] представлення шляху цього графу.

Граф G можливо представити комбінаціями його шляхів. Введемо символ (\vee) як двомісну операцію «або». Так, наприклад, графи зображені на рис. 1 можна записати: $(e_{1,1,2}) \vee (e_{2,1,2})$ – рис. 1 а), $(e_{1,1,2}, e_{2,2,3}) \vee (e_{1,1,2}, e_{3,2,4})$ – рис. 1 б) і $(e_{1,1,2}, e_{2,2,3}) \vee (e_{3,4,2}, e_{2,2,3})$ – рис. 1 в).

В загалі, якщо $\{P_i; i \in K\}$ – множина всіх можливих шляхи графу G , то цей граф представимо некласично, за формулою

$$\tilde{G} \triangleq \bigvee_{i \in K} P_i. \quad (2)$$

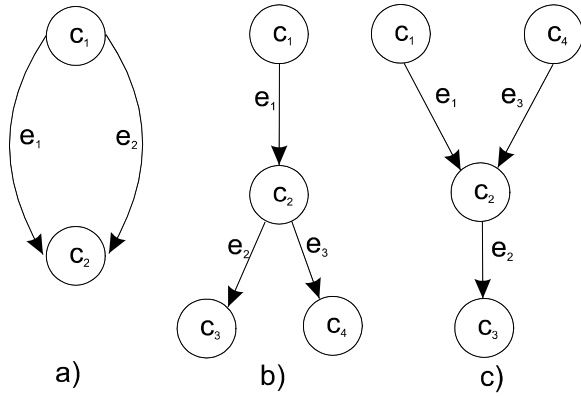


Рис. 1. Приклади розгалужень і злиття шляхів в графах

Від формульного запису (2) графу через його шляхи завжди можливо перейти до класичного представлення графу G . Тому для некласичних графів в подальшому збережемо позначку класичного графу G (виключаємо позначку \sim у графі \tilde{G}). Граф, що складається тільки з одного елементу позначення $G^* \triangleq (e_{k,i,j})$ назовемо *простим*. Очевидно, петлі на вершині графу c_i відповідає граф $G_i^* = (e_{k,i,i})$, який також є простим за формою запису.

Послідовність елементів шляху (1) *конструктивно породжує ланцюжок*

$$l \triangleq e_{s,q,k}, e_{j,k,g}, \dots, e_{r,n,m}$$

та *технологічний ланцюжок*

$$l' \triangleq e_s e_j \dots e_r$$

визначений на алфавітові E . Зрозуміло, що відношення ρ , за яким шляхові (1) ставиться у відповідність технологічний ланцюжок l' однозначне. Але обернене відношення ρ^{-1} не є однозначним – воно тільки вказує на те, що суміжній парі елементів ланцюжка l' ставиться у відповідність суміжні дуги, визначені на довільних позначках вершин. Конструктивний об'єкт, який утворюється технологічним ланцюжком $l' = e_s e_j \dots e_r$ будемо називати *шляховою структурою*

$$\bar{P} \triangleq (e_s, e_j, \dots, e_r)$$

шляху P . Отже, шляхова структура представляється *списком* [6], тобто множиною з фіксованим порядком слідування елементів. Символи у списку можуть повторюватися. Окремої позначки для списку не вводимо, застосовуючи для нього позначки відповідних шляхових структур, але при необхідності список будемо

записувати у вигляді (e_s, e_j, \dots, e_r) , в якому деякі з символів можуть бути порожніми – ε . Порожні символи у списку визначають місце не порожніх символів в упорядкованій множині. Наявність порожнього символу ε у списку відрізняє список від відповідної шляхової структури. Список, що складається тільки з символів $\varepsilon \in E$ є порожнім, тобто $(\varepsilon, \varepsilon, \dots, \varepsilon) = \emptyset$. Список назовемо *нелінійним*, якщо хоча б один його символ $e_q \neq \varepsilon$ у списку повторюється декілька разів підряд. Наприклад, список (a, b, a, b, c) з символів алфавіту E є лінійний, але список (b, a, a, b, c) – нелінійний.

Аналогічно до шляхів введемо поняття *графової структури* \bar{G} графу представленою формулою (2). Графова структура \bar{G} представляється списками шляхів графу G .

Граф G за сукупністю шляхів породжує множину конструктивних ланцюжків $\{l_i\}$ – *мову зв'язаного графу*

$$L(G) \triangleq \{l_i\}$$

та *технологічну мову*

$$L(E) \triangleq \{l'_i\}.$$

Для зв'язаного графу G його графова структура \bar{G} також зв'язана. Відношення між графом G , його графовою структурою \bar{G} і породженими мовами можна зобразити діаграмою

$$\begin{array}{ccc} G & \xrightarrow{\Phi} & L(G) \\ \mu \downarrow & & \downarrow \eta \\ \bar{G} & \xleftarrow{\Psi} & L(E) \end{array} \quad (3)$$

де Φ і Ψ ізоморфні відношення, μ і η гомоморфізми.

Наведена діаграма є комутативною, тому що $\eta \circ \Phi = \Psi^{-1}$, або $\mu = \Psi \circ \eta \circ \Phi$. Це означає еквівалентність відношень μ і η , тобто, за графом G і його мовою $L(G)$ однозначно визначеною утворюються відповідні \bar{G} та $L(E)$, але обернені відтворення не однозначні. Тому для зв'язаної графової структури може існувати множина зв'язаних графів.

Два графи будемо вважати *технологічно еквівалентними*, якщо вони породжують одну і ту ж технологічну мову. Очевидно, графи, які мають однакову графову структуру технологічно еквівалентні.

Введемо деякі формули перетворення графів, які не порушують їх технологічної еквівалентності. Для спрощення запису правил формул перетворень, правила перетворень демонструються на окремих шляхах.

1. Правило скорочення шляху:

$$(e_{s,1,2}, e_{j,2,3}, \dots, e_{r,m-1,m}) = ((e_s e_j \dots e_r)_{1,m}),$$

якому відповідає ланцюжок $l = (e_s e_j \dots e_r)_{1,m}$.

2. Правило виключення паралельної дуги:

$$e_{1,i,j} \vee e_{2,i,j} = (e_1 \vee e_2)_{i,j},$$

при якому вершини зберігаються. Якщо $i = j$, то маємо частковий випадок правила – виключення паралельних петель.

3. Правило виключення альтернативної дуги:

$$e_{1,i,j} \vee e_{2,i,k} = \begin{cases} (e_1 \vee e_2)_{i,j}, & \text{вершина } k \text{ – виключена;} \\ (e_1 \vee e_2)_{i,k}, & \text{вершина } j \text{ – виключена;} \end{cases}$$

за яким дві суміжні дуги замінюються однією – (c_i, c_j) або (c_i, c_k) .

4. Правило виключення петлі:

$$(e_{1,i,i}, (e_{2,i,j} \vee e_{3,i,k})) = (e_1^n e_2)_{i,j} \vee (e_1^n e_3)_{i,k},$$

де e_1^n – технологічний ланцюжок довільної довжини утворений петлею e_{1ii} .

Очевидно, це правило поширюється і на замкнуту петлю альтернативних шляхів, тобто

$$((e_{1,i,k} \vee e_{2,j,k}), e_{3,k,k}) = (e_1 e_3^n)_{i,k} \vee (e_2 e_3^n)_{j,k}.$$

5. Правило виключення контурів:

$$(e_{1,i,j}, e_{3,j,i}) \vee e_{2,i,k} = ((e_1 e_3)^n e_1)_{i,j} \vee ((e_1 e_3)^n e_2)_{i,k},$$

яке узагальнює правило 4.

6. Правило виключення вершини:

$$(e_{1,i,m}, e_{3,m,n}) \vee (e_{2,j,m}, e_{4,m,k}) = ((e_1 e_4)_{i,k} \vee (e_1 e_3)_{i,n}) \vee ((e_2 e_3)_{j,n} \vee (e_2 e_4)_{j,k}).$$

Для створення графових композицій введемо операції над графами.

Нехай $\{l_{1,i}\}$ і $\{l_{2,j}\}$ мови породжені зв'язаними графами G_1 і G_2 . Граф G_1 є підграфом графу G_2 за операцією включення (\prec), тобто $G_1 \prec G_2$, якщо множина $\{l_{1,i}\}$ є підмножиною множини $\{l_{2,j}\}$. Відповідно, для графових структур \bar{G}_1 і \bar{G}_2 , утворених технологіч-

ними ланцюжками $\{l'_{1,i}\}$ та $\{l'_{2,j}\}$, включення $\bar{G}_1 \prec \bar{G}_2$ означає, що $\{l'_{1,i}\} \subset \{l'_{2,j}\}$ і \bar{G}_1 є графовою підструктурою структури \bar{G}_2 . Для окремих технологічних ланцюжків l'_1 і l'_2 під включенням $l'_1 \subset l'_2$ розуміється по символівне включення за порядком слідування елементів ланцюжків.

Схожість введеного включення над графом G і його структурою \bar{G} є зовнішня, але за результатами вони різні: з включення $G_1 \prec G_2$ обов'язково має місце $\bar{G}_1 \prec \bar{G}_2$. Крім того, з наявності включення $\bar{G}_1 \prec \bar{G}_2$ за діаграмою (2) не обов'язково існує включення $G_1 \prec G_2$. Наприклад, якщо графи G_1 і G_2 породжують тільки ланцюжки $l_1 = a_{1,2}, b_{2,2}, a_{2,3}$ та $l_2 = a_{1,2}, a_{2,2}, c_{2,3}, b_{3,4}, a_{4,4}$, то між цими графами не існує відношення за включенням, але між відповідними графовими структурами $\bar{G}_1 = (a, b, a)$ і $\bar{G}_2 = (a, a, c, b, a)$, як за списками виконується по символівне включення $\bar{G}_1 \prec \bar{G}_2$ по першому, четвертому та п'ятому відповідним елементам списку \bar{G}_2 або по другому, четвертому та п'ятому елементам цього ж списку. Нагадуємо, що для списків послідовність елементів суттєва. Отже, включення $\bar{G}_1 \prec \bar{G}_2$ неоднозначне, оскільки воно характеризується двома структурами включень (a_1, b_4, a_5) і (a_2, b_4, a_5) . Таким чином, для існування включення (\prec) між графовими структурами \bar{G}_1 і \bar{G}_2 необхідно і достатньо виконання умов:

– кількість елементів графової структури \bar{G}_1 не повинна бути більшою за кількість елементів структури \bar{G}_2 ;

– кількість кожного символу (з врахуванням його повторень) графової структури \bar{G}_1 повинна бути не більшою ніж кількості цих же символів в структурі \bar{G}_2 ;

– послідовність розташування символів графової структури \bar{G}_1 не повинна порушуватися хоча б в одній послідовній комбінації тих же символів графової структури \bar{G}_2 .

За допомогою введеної операції включення можна встановити певний порядок на множенні підграфів $\{G_i\}$ деякого графу G . Так, підграф G_k графу G на множині підграфів $\{G_i\}$ є *максимальним за включенням*, якщо у цій множині

можна вказати таку підмножину $\{G_j; j \in K\}$, що для всіх її елементів має місце ланцюг за включенням $G_{j_1} \prec G_{j_2} \prec \dots \prec G_{j_m} = G_k$ і серед елементів множини $\{G_i\}$ не існує такого $- G_s$, щоб $G_k \prec G_s$. З іншого боку підграф G_r графу G на множині підграфів $\{G_i\}$ є мінімальним за включенням, якщо у цій множині можна вказати таку підмножину $\{G_s; s \in N\}$, що для її елементів має місце ланцюг за включенням типу $G_r = G_{j_1} \prec G_{j_2} \prec \dots \prec G_{j_m}$, і серед елементів множини $\{G_i\}$ не існує такого $- G_q$, щоб $G_q \prec G_r$. Таким же чином через визначення максимальної $\max_{\prec, i} \{\bar{G}_i\}$ і мінімальної $\min_{\prec, i} \{\bar{G}_i\}$ структур за включеннями можна ввести порядок на множині підструктур $\{\bar{G}_i\}$ графової структури \bar{G} графу G .

Під операцією перетину (\cap) двох підграфів G_1 і G_2 графу G розуміється їх спільний підграф (підграфи), інакше $G_1 \cap G_2 \neq \emptyset$. Так для підграфів

$$G_4 = (a_{1,2}, b_{2,3}, c_{3,5}, e_{5,2}, b_{2,3}, c_{3,5}, a_{5,6}),$$

$$G_5 = (a_{1,2}, b_{2,4}, a_{4,4}^2, b_{4,5}, a_{5,6}); \quad (4)$$

результатом перетину $G_4 \cap G_5$ є два їх підграфи $G_1^* = (a_{1,2})$ та $G_2^* = (a_{5,6})$.

Операцію перетину графових підструктур (списків) \bar{G}_1 і \bar{G}_2 структури \bar{G} позначимо як $\bar{G}_1 \cap \bar{G}_2$. Результат перетину є графова підструктура (підструктури) \bar{G}_3 , яка знаходиться наступним чином.

1. З двох списків \bar{G}_1 і \bar{G}_2 вибираємо список з меншою кількістю елементів, нехай це буде, для визначеності, графова структура \bar{G}_2 .

2. Формується графова підструктура $\bar{G}_{3,1}$ така, що $\bar{G}_{3,1} \prec \bar{G}_1$ і $\bar{G}_{3,1} \prec \bar{G}_2$:

– починаючи з першого символу e_1 структури \bar{G}_2 , перевіряємо його першу, за порядком слідування, наявність у списку \bar{G}_1 , або наступну наявність. Нехай це буде символ e_i , тоді символ e_1 приймається за перший елемент списку $\bar{G}_{3,1}$. Якщо символ e_1 у списку \bar{G}_1 відсутній, то до списку $\bar{G}_{3,1}$ заносимо порожній символ ε . Зауважимо, що при наявності альтерна-

тивних шляхів у графові G_1 вище визначена перевірка виконується для кожного списку графової структури \bar{G}_1 ;

– перевіряємо наявність другого елементу $e_2 \in \bar{G}_2$ у списку \bar{G}_1 після символу e_i , якщо попередній символ $e_1 \notin \bar{G}_1$, то перевіряється наявність символу e_2 спочатку списку \bar{G}_1 або його наступну наявність; при виконанні умови $e_2 \in \bar{G}_1$ за другий елемент списку $\bar{G}_{3,1}$ приймається e_2 , в іншому випадку маємо порожній символ ε ;

– процес перевірки наявності елементів графової структури \bar{G}_2 у списку \bar{G}_1 продовжується поки не вичерпаються елементи списку (списків) \bar{G}_2 , зрозуміло, що для так побудованого списку $\bar{G}_{3,1}$ включення $\bar{G}_{3,1} \prec \bar{G}_1$ і $\bar{G}_{3,1} \prec \bar{G}_2$ виконуються.

3. За методикою пункту 2, починаючи з символу $e_2 \in \bar{G}_2$, формується підструктура $\bar{G}_{3,2}$ така, що $\bar{G}_{3,2} \prec \bar{G}_1$ і $\bar{G}_{3,2} \prec \bar{G}_2$:

4. Побудова наступних підструктур $\bar{G}_{3,j}$ продовжується, поки не будуть перебрані всі елементи зі структури \bar{G}_2 .

5. Серед множини графових підструктур $\{\bar{G}_{3,j}\}$ знаходимо максимальні за включенням підструктури, які приймаємо за результат перетину \bar{G}_3 , тобто $\bar{G}_1 \cap \bar{G}_2 = \max_{\prec, j} \{\bar{G}_{3,j}\}$; якщо елементи множини $\{\bar{G}_{3,j}\}$ утворені тільки порожніми списками, то за результат перетину приймаємо $\bar{G}_1 \cap \bar{G}_2 = \emptyset$.

Для прикладу застосування операції перетину списків розглянемо наведені вище графи (4), за якими

$$\bar{G}_4 = (a, b, c, e, b, c, a) \text{ та } \bar{G}_5 = (a, b, a, a, b, a).$$

Список \bar{G}_5 менший ніж \bar{G}_4 , тому перевіряємо наявність елементів структури \bar{G}_5 у структурі \bar{G}_4 . Перший символ $a \in \bar{G}_5$ є першим у списку \bar{G}_4 з чого маємо, що $a \in \bar{G}_{3,1}$ і цей символ є першим елементом списку $\bar{G}_{3,1}$. Аналогічно другий символ $b \in \bar{G}_5$ є другим елементом списку \bar{G}_4 , тому список $(a, b) \prec \bar{G}_{3,1}$. Наступний символ $a \in \bar{G}_5$ міститься на сьомому місці спи-

ску \bar{G}_4 і тому маємо $(a,b,a) \prec \bar{G}_{3,1}$. Далі символи графу \bar{G}_5 повинні міститися за сьомим елементом списку \bar{G}_4 . Отже, отримаємо список $\bar{G}_{3,1} = (a,b,a,\varepsilon,\varepsilon,\varepsilon)$. Таким же чином знаходиться список $\bar{G}_{3,2} = (b,a,\varepsilon,\varepsilon,\varepsilon)$. Продовжуючи побудову інших списків $\bar{G}_{3,j}$, отримаємо множину списків

$$\{(a,b,a,\varepsilon,\varepsilon,\varepsilon), (b,a,\varepsilon,\varepsilon,\varepsilon), (a,a,\varepsilon,\varepsilon), (a,b,a), (b,a), (a)\}.$$

Нескладно бачити, що максимальними за включенням списками на цій множині є списки $(a,b,a,\varepsilon,\varepsilon,\varepsilon)$ і $(a,a,\varepsilon,\varepsilon)$, тобто перетин графових структур є $\bar{G}_4 \cap \bar{G}_5 = \{(a,b,a), (a,a)\}$. Доцільно перевірити результат перетину на зв'язаність графових структур. Для цього спочатку з'ясуємо, які структури включень має графова підструктура $\bar{G}_{3,1} = (a,b,a)$. Структура $\bar{G}_{3,1}$ характеризується наступною множиною структур включень $\{(a_1,b_2,a_7), (a_1,b_5,a_7)\}$, що означає фіксованість місць для символу a , але місце символу b не визначене. Подальший аналіз графових структур \bar{G}_4 і \bar{G}_5 не розв'язує цього питання, а також питання зв'язаності комбінацій символів ab та ba . Тому для підструктури $\bar{G}_{3,1}$ її зв'язаність не визначена. Аналіз підструктури $\bar{G}_{3,3} = (a,a)$ показує, що вона характеризується тільки однією структурою включення, отже, підструктура $\bar{G}_{3,3}$ зв'язана. Таким чином, перетин графових структур $\bar{G}_4 \cap \bar{G}_5$ має зв'язану невизначеність. Розв'язати цю невизначеність можливо при наявності інформації про інші графові підструктури структури \bar{G} .

За різницю (\dashv) двох підграфів G_1 і G_2 графу G таких, що $G_1 \cap G_2 \neq \emptyset$, приймемо граф (графи) $G_1 \dashv G_2$, визначений як граф G_1 з видаленими спільними дугами графів G_1 і G_2 .

Розповсюдимо операцію різниці на списки \bar{G}_1 і \bar{G}_2 [6]. Під різницею $\bar{G}_1 \dashv \bar{G}_2$ будемо розуміти такий список, який складається з елементів списку \bar{G}_1 , котрі відсутні у списку \bar{G}_2 , а замість видалених елементів перетину $\bar{G}_1 \cap \bar{G}_2$ записуються порожні символи.

Наприклад, для графів (4) різниця $G_4 \dashv G_5 = (b_{2,3}, c_{3,5}, e_{5,2}, b_{2,3}, c_{3,5})$, але різницею між списками $\bar{G}_4 \dashv \bar{G}_5$ буде список $(\varepsilon, \varepsilon, c, e, \varepsilon, c, \varepsilon)$, що відповідає графовій структурі (c, e, c) .

Операція об'єднання (\cup) підграфів G_1 і G_2 графу $G \in G_1 \cup G_2$ така, що отримаємо граф G_3 , якщо $G_1 \cap G_2 \neq \emptyset$ або графи G_1 і G_2 , якщо $G_1 \cap G_2 = \emptyset$. Граф $G_3 = G_1 \cup G_2$ можна легко знайти, якщо підграфи G_1 і G_2 мають спільну вершину або спільну дугу. В інших випадках їх об'єднання знаходиться складніше. Наприклад, коли графи $G_1 = (a_{1,2}, b_{2,3})$, $G_2 = (c_{4,2}, d_{2,5})$ мають спільну вершину з позначкою «2», тоді

$$G_1 \cup G_2 = ((a_{1,2}, b_{2,3}) \vee (c_{4,2}, d_{2,5}) \vee (a_{1,2}, d_{2,5}) \vee (c_{4,2}, b_{2,3})),$$

або після застосування третього правила перетворення графів цей вираз спрощується –

$$G_1 \cup G_2 = ((a \vee c)_{1,2}, (b \vee d)_{2,3}).$$

Для іншого прикладу, якщо у графів $G_1 = (a_{1,2}, c_{2,4})$ і $G_2 = (b_{3,2}, c_{2,4})$ спільна дуга $c_{2,4}$, то

$$G_1 \cup G_2 = ((a_{1,2} \vee b_{3,2}), c_{2,4}) = ((a \vee b)_{1,2}, c_{2,4}).$$

Операцію об'єднання застосуємо і для списків. Під об'єднанням списків $\bar{G}_1 \cup \bar{G}_2$ будемо розуміти список як конкатенацію списків \bar{G}_1 і \bar{G}_2 . Так для $\bar{G}_1 = (a,b)$ і $\bar{G}_2 = (c,d)$ їх об'єднання є $\bar{G}_1 \cup \bar{G}_2 = (a,b,c,d)$. Слід звернути увагу на те, що операція об'єднання списків не комутативна.

Розглянемо множину всіх підграфів $\{G_i\}$ графу G , яка містить і власний підграф G . За відношенням включення (\prec) для будь якого підграфу $G_k \in \{G_i\}$ існує елемент $G_j \in \{G_i\}$ такий, що має місце $G_j \prec G_k$ або $G_k \prec G_j$. Отже, множина $\{G_i\}$ і будь яка її підмножина з підграфом G за цим відношенням може бути розбита на такі класи $\{K_r; r \in R\}$, що елементи $G_s^r \in K_r$, $s = \overline{1, m_r}$ пов'язані ланцюгом за включенням $G_1^r \prec G_2^r \prec \dots \prec G_{m_r}^r$; $G_q^r \in \{G_i\}$. Вочевидь, кожен з класів K_r може містити підграфи з других класів. Так кожен клас має один спільний максимальний за включенням граф

G , але один можливо унікальний мінімальний за включенням підграф. Зрозуміло, що і множину графових структур $\{\bar{G}_i\}$ таким же чином можна розбити на класи $\{\bar{K}_v; v \in V\}$.

Визначення 1. Сукупність різних мінімальних за включенням підграфів класів K_r розбиття множини $\{G_j\} \subseteq \{G_i\}$ називається *базисом* цієї множини. Відповідно, сукупність мінімальних за включенням графових підструктур класів \bar{K}_r розбиття підмножин $\{\bar{G}_j\} \subseteq \{\bar{G}_i\}$ утворює *графовий структурний базис* цієї підмножини.

Якщо прийняти підграфи ланцюга за включенням деякого класу K_r за вершини, а відношення (\prec), яким зв'язані підграфи цього ланцюга, за дуги, то отримуємо деякий графовий лінійний шлях P_r на класі K_r . За початкову вершину шляху приймемо максимальний за включенням граф G .

Визначення 2. Зв'язану сукупність шляхів P_r множини класів $\{K_r; r \in R\}$ ($\{\bar{K}_v; v \in V\}$) зі спільною вершиною G або \bar{G} , назовемо *структурним графом* відповідних множин підграфів (графових підструктур).

Наприклад, множина підграфів $M_G = \{G_1, G_2, G_3, G_4, G_5, G\}$ графу G ланцюгами $G_1 \prec G_3 \prec G_5 \prec G$ і $G_2 \prec G_4 \prec G$ розбивається на класи: $K_1 = \{G_1, G_3, G_5, G\}$, $K_2 = \{G_2, G_4, G\}$. Структурний граф множини M_G зображений на рис. 2.

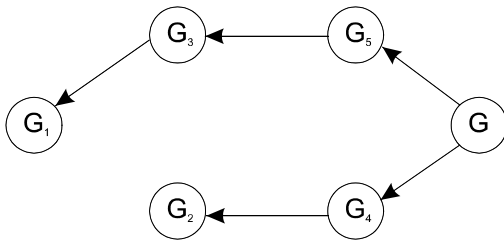


Рис. 2. Приклад структурного графу

Заключні вершини структурного графу множини $\{G_i\}$ утворюють її базис, так для графу на рис. 2 базис множини M_G утворюють графи G_1 і G_2 .

Визначення 3. Сукупність S_G підграфів G_j графу G , таких, що $\{G_j; \cup_j G_j = G\}$ назовемо *системою утворюючих підграфів графу G* .

Система S_G назовемо *повною системою*, якщо в ній немає таких підграфів, видалення яких не впливає на утворення графу G .

Визначення 4. Система утворюючих підграфів графу G називається *мінімальною S_G^0* , якщо вона повна і створена на базисі множини підграфів графу G , або на комбінаціях базису з деякими підграфами графу G за допомогою операцій: (\cap), (\cup) і (\rightarrow).

Тепер перейдемо до розгляду задачі відтворення графу G за деякими відомими зразками технологічної мови $\{l'_i\} \subset L(E)$.

Відтворення графів за зразками мови $L(E)$

Задача. Нехай задані зразок

$$Z = \{l'_i; i \in I, l'_i \in L(E)\}$$

і відповідна сукупність його графових структур $S = \{\bar{G}_i\}$. Необхідно на парі $\langle Z, S \rangle$ відтворити такий орієнтований навантажений символами алфавіту E граф G , щоб породжена цим графом технологічна мова L_0 містила в собі заданий зразок.

Зрозуміло, що так поставлена задача відтворення графу за висновками з діаграми (3) розв'язується не однозначно. Для часткового усунення неоднозначності розв'язку задачі введемо деякі припущення:

- 1) рішення відносно технологічної мови графу приймаються зовні задачі;
- 2) рішення відносно кількості елементів та повноти технологічного зразка приймаються поза межами задачі;
- 3) ланцюжкам зразка відповідають однозначні графові структури;
- 4) припущення відносно початкових і заключних станів графу приймаються за межами задачі;
- 5) графові структури ланцюжків зразка створюють деяку систему графових підструктур відтворюючого графу.

Перейдемо до розгляду алгоритму за яким пропонується розв'язувати наведену задачу.

Розв'язувати задачу будемо за допомогою побудови утворюючої системи підграфів. Схема розв'язку задачі спирається на введені припущення і результати попереднього пункту, тому не вдаючись до деталей наведемо її узагальнений алгоритм:

- 1) за пунктами 4) і 5) припущень провести аналіз технологічного зразка та його графових підструктур і виконати їх спрощення, переозначення елементів списків, тощо; на перетворених графових структурах створити систему S ;

2) на системі S побудувати структурний граф залежностей за включенням (за визначенням 2);

3) на заключних вершинах структурного графу виділити базис (за визначенням 1) системи S і на ньому знайти максимальні за включенням лінійні графові підструктури \bar{G}_i , застосовуючи при необхідності операції (\cap, \supset) ;

4) за лінійними підструктурами \bar{G}_i , з'ясовуючи зв'язаність їх елементів, відтворити графи G_i ;

5) знайдені у пункті 4) лінійні підграфи об'єднати у лінійний кістяк [5] $\{G_i\}$ невідомого графу G , який включити до мінімальної утворюючої системи S_G^0 (за визначенням 4);

6) за допомогою багаторазового застосування операцій (\cup, \cap, \supset) над елементами системи S , її базису і кістяку виділити всі можливі прості підграфи G_j^* (відсутні у лінійному кістякові);

7) доповнити утворюючу систему S_G^0 простими підграфами G_j^* ;

8) на мінімальній системі підграфів S_G^0 побудувати відтворений граф;

9) за допомогою правил перетворення графів 1 – 6 побудувати еквівалентний до відтвореного графу G простий граф G^* ;

10) позначку простого графу G^* прийняти за технологічну мову L_0 відтвореного графу, переконатися у тому, що $Z \subset L_0$.

Для прикладу, розглянемо деякий технологічний процес, який складається з елементарних технологічних операцій $\{a, b, c, d, e\}$, які приймемо за елементи алфавіту E . Нехай також відомі декілька технологій цього процесу, які, для визначеності, починаються з початку і завершуються в кінці процесу. Представимо їх у вигляді наступних технологічних ланцюжків:

$$\begin{aligned} l_1^t &= ab^3ca^3ba, \quad l_2^t = a^2b^3ad^2, \\ l_3^t &= a^3b^3ad^4ea^2b^3ca^4badeaba, \quad l_4^t = adea^2c, \\ l_5^t &= a^2b^2dea^2ceabad, \quad l_6^t = a^2b^2dedcadea^3c, \\ l_7^t &= abdbd^2ea^2dec, \quad l_8^t = abdeadadeabad, \end{aligned} \quad (5)$$

де у виразах (5) степінь k над символом означає кількість повторень цього символу в технологічному ланцюжкові.

Приймемо ланцюжки (5) за множину зразка Z . Будемо вважати, що кожен з ланцюжків l_i^t зразка Z задає деяку графову структуру \bar{G}_i ; $i = \overline{1,8}$ технологічного процесу так, що їх початкові та заключні стани співпадають із відповідними станами поки невідомого графу G . Нехай ці підструктури представлені виразами:

$$\begin{aligned} \bar{G}_1 &= (a, b, b, b, c, a, a, a, b, a), \\ \bar{G}_2 &= (a, a, b, b, b, a, d, d), \\ \bar{G}_3 &= (a, a, a, b, b, b, a, d, d, d, d, e, a, a, b, b, b, c, a, a, a, b, a, d, e, a, b, a), \\ \bar{G}_4 &= (a, d, e, a, a, c), \\ \bar{G}_5 &= (a, a, b, b, d, e, a, a, c, e, a, b, a, d), \\ \bar{G}_6 &= (a, a, b, d, e, d, c, a, d, e, a, a, c), \\ \bar{G}_7 &= (a, b, d, b, d, d, e, a, a, d, e, c), \\ \bar{G}_8 &= (a, b, d, e, a, d, a, d, e, a, b, a, d). \end{aligned}$$

Аналізуючи зміст ланцюжків зразка (5) та їх графові структури і враховуючи припущення відносно початкового та заключного станів графу, приходимо до висновку, що початковий стан графу G має не меншу ніж один степінь з вихідною дугою a , а заключному стану відповідає вершина степені не менше ніж три з трьома входами a, d, c . Приймемо початкову вершину графу G за 1, а заключну за невідоме k . Отже з урахуванням цього аналізу за пунктом 1) алгоритму отримаємо систему S , елементами якої будуть підструктури \bar{G}_i ; $i = \overline{1,8}$.

Тепер за другим пунктом алгоритму на системі S побудуємо структурний граф, який зображено на рис. 3.

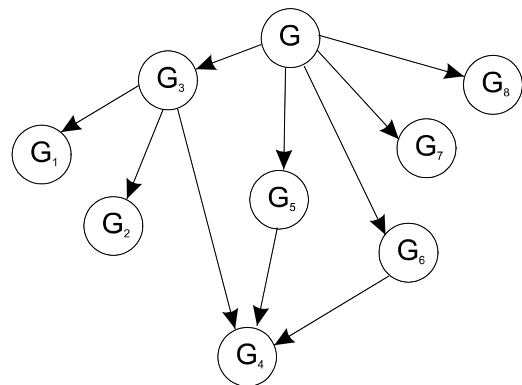


Рис. 3. Структурний граф залежностей елементів системи S

Як видно, із структурного графу рис. 3, базис графу G складається з підграфів $\bar{G}_1, \bar{G}_2, \bar{G}_4, \bar{G}_7$ і \bar{G}_8 множини $\{\bar{G}_i; i = \overline{1,8}\}$, тому

за наступним пунктом алгоритму на цьому базисі знайдемо лінійні підструктури. Для цього спочатку розглянемо перетин графових підструктур \bar{G}_1 та \bar{G}_2 і знайдемо на множині списків, з якої визначається перетин $\bar{G}_1 \cap \bar{G}_2$, максимальну за включенням лінійну підструктуру $\bar{G}_{1,0}$. Перетин структур \bar{G}_1 і \bar{G}_2 визначається за правилом

$$\begin{aligned} \bar{G}_1 \cap \bar{G}_2 = \max \{ & (a^2, b, \varepsilon^2, a, \varepsilon^2), (a^2, b^3, a, \varepsilon^2), \\ & (a, b, \varepsilon^2, a, \varepsilon^2), (b^3, a, \varepsilon^2), (b^3, a, \varepsilon^2), (b, a, \varepsilon^2), \\ & (a, \varepsilon, \varepsilon), (\varepsilon, \varepsilon), (\varepsilon) \}, \end{aligned}$$

тому лінійним максимальним за включенням списком даної множини є $(a, b, \varepsilon, \varepsilon, a, \varepsilon, \varepsilon)$.

Отже, лінійна підструктура $\bar{G}_{1,0}$ відповідає технологічному ланцюжкові $l'_{1,0} = aba$ породженому невідомим графом $G_{1,0}$. З'ясуємо зв'язковість списку $\bar{G}_{1,0}$. При побудові множини списку перетину $\bar{G}_1 \cap \bar{G}_2$ список $(a, b, \varepsilon, \varepsilon, a, \varepsilon, \varepsilon)$ визначається неоднозначно. Так графова структура $\bar{G}_{1,0} = (a, b, a)$ визначається різними структурами включень, наприклад, (a_1, b_9, a_{10}) і (a_8, b_9, a_{10}) , в яких індекси вказують на номер відповідного символу в графовій структурі \bar{G}_1 . Причому всі структури включення для графу $\bar{G}_{1,0}$ вказують на однаковий зв'язок суміжних символів b і a . Аналіз графових структур \bar{G}_1 і \bar{G}_2 показує, що символи a і b в послідовності (a, b) також зв'язані. Тому граф $G_{1,0}$ зв'язаний і визначений на чотирьох вершинах, причому його початкова вершина співпадає з початковою вершиною графу G . Позначимо ці вершини символами 1, 2, 3 і 4, таким чином отримуємо граф

$$G_{1,0} = (a_{1,2}, b_{2,3}, a_{3,4}). \quad (6)$$

Залишається не з'ясованим питання, чи є вершина 4 графу G заключною. Відповідь на це питання знайдемо, розглянувши різницю $\bar{G}_2 \rightarrow \bar{G}_{1,0}$, яка має, на заключній вершині графу G петлю з позначкою d . Тоді перетин $\bar{G}_{1,0} \cap (\bar{G}_2 \rightarrow (d, d))$ вказує на те, що заключні вершини графів $G_{1,0}$ і G однакові. Отже визначено, що $k = 4$.

Розглянемо тепер перетин $\bar{G}_7 \cap \bar{G}_4$, якому відповідає

$$\max \{ (a, d, e, a, a, c), (a, d, e, \varepsilon, \varepsilon, c), (d, e, a, a, c),$$

$$(d, e, \varepsilon^2, c), (e, a, a, c), (e, \varepsilon, \varepsilon, c), (a, a, c), (a, c), (c) \}.$$

Для цієї множини лінійним максимальним за включенням буде список $(a, d, e, \varepsilon, \varepsilon, c)$, котрий визначає графову структуру $\bar{G}_{2,0} = (a, d, e, c)$. З'ясуємо зв'язаність структури $\bar{G}_{2,0}$. Множина його структур включень має вигляд

$$\{ (a_1, d_3, e_{11}, c_{12}), (a_1, d_5, e_{11}, c_{12}), (a_1, d_6, e_{11}, c_{12}), \\ (a_1, d_{10}, e_{11}, c_{12}), (a_8, d_{10}, e_{11}, c_{12}), (a_9, d_{10}, e_{11}, c_{12}) \}.$$

Аналіз множини структур включень показує, що послідовність e, c зв'язана. З аналізу графових структур \bar{G}_7 і \bar{G}_4 також маємо, що послідовність d, e є зв'язаною. Зв'язаність символів a і d зостається поки що не визначеною. Тому граф $G_{2,0}$ визначений на п'ятьох або шістьох вершинах, з яких початкова і заключна співпадають з відповідними вершинами графу G . Для визначеного зв'язку між дугами d, e, c графу $G_{2,0}$ введемо дві нові позначки вершин 5 і 6, тоді можна записати

$$G_{2,0} = (a_{1,x}, d_{y,5}, e_{5,6}, c_{6,4}), \quad (7)$$

де x, y – поки невизначені вершини.

Перетин підструктури \bar{G}_8 з іншими елементами базису не утворює нових лінійних графових підструктур, які б починалися в початковій і завершувалися в заключній вершинах графу G . Таким чином, повний базис повинен складатися тільки з підструктур $\bar{G}_1, \bar{G}_2, \bar{G}_4, \bar{G}_7$. Отже, кістяком (зі спільною заключною вершиною) утворюючої системи S_G^0 буде множина елементів (6) і (7), тобто $\{G_{1,0}, G_{2,0}\}$.

Знайдемо тепер всі прості підграфи, які відсутні у кістякові. Аналіз різниць $\bar{G}_1 \rightarrow \bar{G}_{1,0}$ та $\bar{G}_2 \rightarrow \bar{G}_{1,0}$ і структур \bar{G}_1, \bar{G}_2 показує, що на вершинах з позначками 1, 2 і 4 маються петлі $a_{1,1}, b_{2,2}$ і $d_{4,4}$, крім того перша різниця з врахуванням петель $a_{1,1}$ і $b_{2,2}$ показує наявність контура $c_{3,1}$ в підграфові G_1 . Отриманої інформації достатньо для відтворення підграфів G_1 і G_2 , котрі породжують ланцюжки зразка l'_1 і l'_2 .

Так об'єднуючи лінійний граф (6) з контуром $(c_{3,1})$ і враховуючи те, що цей контур охоплює дуги $a_{1,2}$ та $b_{2,3}$, отримаємо граф $G_{1,0} \cup (c_{3,1}) = G_{3,0}$,

$$G_{3,0} = (a_{1,2}, b_{2,3}, c_{3,1}, a_{1,2}, b_{2,3}, a_{3,4}). \quad (8)$$

Об'єднання графу (8) з відповідними кількостями петель $(b_{2,2})$, $(a_{1,1})$ відтворює граф $G_1 = (a_{1,2}, b_{2,2}, b_{2,3}, c_{3,1}, a_{1,1}, a_{1,2}, b_{2,3}, a_{3,4})$, а об'єднання графу (6) з петлями $a_{1,1}$, $b_{2,2}$ і $d_{4,4}$ відтворює граф $G_2 = (a_{1,1}, a_{1,2}, b_{2,2}, b_{2,3}, a_{3,4}, d_{4,4})$.

На графах (6) і (8) та простих графах $(a_{1,1})$, $(b_{2,2})$, $(d_{4,4})$ за допомогою операції об'єднання також можна побудувати нові підграфи, наприклад, підграфи, які породжують підланцюжки $l'_9 = a^3 b^3 a d^4$, $l'_{10} = a^2 b^3 c a^4 b a d$ ланцюжка зразка l'_3 . Тому за різницею $\bar{G}_3 \ominus \bar{G}_9 \ominus \bar{G}_{10} \ominus \bar{G}_{1,0}$ зясуємо наявність у графові G контуру $e_{4,1}$. Інших утворюючих підграфів графу G на гілках (G, G_3, G_1) і (G, G_3, G_2) структурного графу (дивися рис. 3) не має. Відтворений підграф (на чотирьох вершинах) графу G зображено на рис. 4

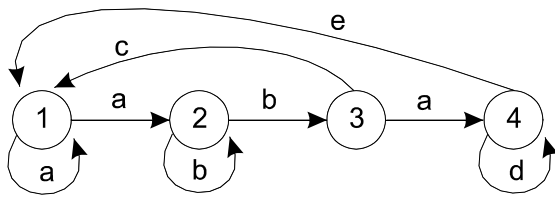


Рис. 4. Відтворений підграф графу G

Тепер за рис. 4 нескладно записати граф

$$G_3 = (a_{1,1}^2, a_{1,2}, b_{2,2}^2, b_{2,3}, a_{3,4}, d_{4,4}^4, e_{4,1}, a_{1,1}, a_{1,2}, b_{2,2}^2, b_{2,3}, c_{3,1}, a_{1,2}, b_{2,3}, a_{3,4}, d_{4,4}, e_{4,1}, a_{1,2}, b_{2,3}, a_{3,4}),$$

котрий породжує технологічний ланцюжок l'_3 .

Продовжимо виконання пункту 5) алгоритму відтворення графу G .

Аналізуючи граф $G_{2,0}$ та графові структури \bar{G}_5 , \bar{G}_6 і стани графу зображеного на рис. 4, встановлюємо, що невизначена у графові $G_{2,0}$ вершина $x = y = 2$. Отже, за формулою (7) маємо другий лінійний утворюючий підграф

$$G_{2,0} = (a_{1,2}, d_{2,5}, e_{5,6}, c_{6,4}).$$

Розглядаючи потім різницю $\bar{G}_4 \ominus \bar{G}_{2,0}$ встановлюємо, що граф G_4 має на вершині 6 петлю $a_{6,6}$, тому $G_4 = (a_{1,2}, d_{2,5}, e_{5,6}, a_{6,6}^2, c_{6,4})$. Далі виконуючи операції (\cup, \ominus) над графовими структурами $\bar{G}_{2,0}$, \bar{G}_4 , \bar{G}_6 , \bar{G}_7 і враховуючи граф $(a_{6,6})$ аналогічно тому, як це робилося вище, визначимо, що граф G додатково має контур $b_{5,4}$ і дугу – контур $d_{5,3}$.

Таким чином, за структурним графом (рис. 3) і операціями (\cup, \cap, \ominus) над деякими його підграфами побудовано множину простих графів

$$Q^* = \left\{ (a_{1,1}), (a_{6,6}), (b_{2,2}), (d_{4,4}), (c_{3,1}), (e_{4,1}), (b_{5,4}), (d_{6,3}) \right\}.$$

Отже, на технологічному зразкові Z побудована мінімальна утворююча система $S_G^0 = \{G_{1,0}, G_{2,0}, Q^*\}$ графу G .

В межах утворюючої системи S_G^0 , як показано вище, відтворені графи: G_1 , G_2 , G_3 і G_4 . Тепер також можна відтворити інші графи G_5 , G_6 , G_7 повної системи S :

$$\begin{aligned} & (a_{1,1}, a_{1,2}, b_{2,2}^2, d_{2,5}, e_{5,6}, a_{6,6}^2, c_{6,4}, e_{4,1}, a_{1,2}, b_{2,3}, a_{3,4}, d_{4,4}), \\ & (a_{1,1}, a_{1,2}, b_{2,2}, d_{2,5}, e_{5,6}, d_{6,3}, c_{3,1}, a_{1,2}, d_{2,5}, e_{5,6}, a_{6,6}^3, c_{6,4}), \\ & (a_{1,2}, b_{2,2}, d_{2,5}, b_{5,4}, d_{4,4}, e_{4,1}, a_{1,1}, a_{1,2}, d_{2,5}, e_{5,6}, c_{6,4}). \end{aligned}$$

Якщо для впорядкування вершин графу G ввести перепозначення вершин в утворюючій системі S_G^0 , замінивши позначки вершин за правилами: $4 \rightarrow 6$, $5 \rightarrow 4$, і $6 \rightarrow 5$, то отримаємо відтворений граф, представлений на рис. 5.

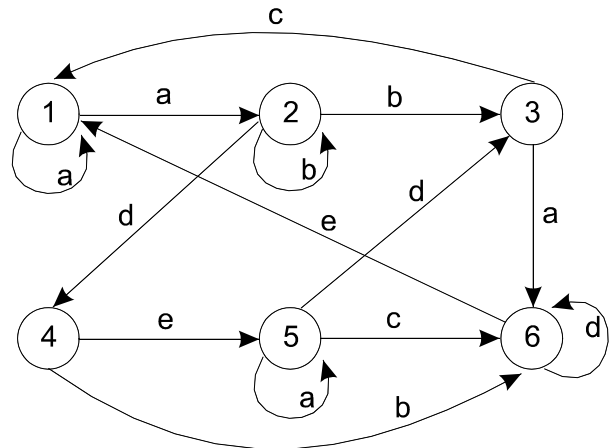


Рис. 5. Відтворений граф

Отриманий нелінійний граф має досить велике цикломатичне число $(v=9)$, тому поро-

джену ним технологічну мову безпосередньо за графом G складно визначити. Для цього скористаємося правилами перетворення графу. За пунктом 10) алгоритму зведемо граф, зображений на рис. 5, до простого шляху за такою послідовністю застосування правил виключення:

- виключимо спочатку всі петлі на вершинах: 1, 2, 5 і 6;
- виключимо вершину 1 графу, замкнувши обернені дуги $c_{3,1}$ і $e_{6,1}$ на вершину 2;
- виключимо вершину 5 і утворені паралельні дуги на вершинах 2 та 3;
- виключимо контур (c_{32}) і тримаємо граф $((TM)_{2,6} \vee (VR)_{2,6}, e_{6,2})$;
- виключимо паралельні дуги і контур $(e_{6,2})$.

Отже, отримаємо технологічну мову

$$L(G) = ((TM | VR)e)^n (TM | VR), \quad (9)$$

де $T = ((A | DQ)c)^n (A | DQ)$, $V = ((A | DQ)c)^n D$,
 $R = ea^n cd^n | bd^n$, $M = bd^n$, $A = a^n b^n$, $D = a^n b^n d$
і $Q = ea^n d$.

Таким чином, за заданим зразком (3) вдалося відтворити граф (рис. 5) і його технологічну мову (9).

Висновки

В результаті виконаних досліджень по розв'язанню проблеми відтворення графів встановлено, що:

- процес відтворення графів спирається на не повну інформацію відносно процесів, технологій і інше, тому передбачає введення деяких припущень;
- розв'язок проблеми не однозначний і вимагає напрацювань спеціальних підходів до її вирішення, особливо у випадку складних

графових моделей представлення предметних областей;

- проблему відтворення графів зручно розв'язувати за допомогою систем утворюючих підграфів, які визначаються за неповною інформацією зразків підграфів;
- розроблена метода згортання графів дозволяє знайти технологічну мову, яка породжується відтвореним графом;
- запропонований алгоритмічний підхід до розв'язання проблеми відтворення графів є досить простим в застосуванні і універсальним;
- за запропонованим алгоритмом відтворення графів можна розробити автоматизовану систему відтворення графових моделей.

БІБЛОГРАФІЧНИЙ СПИСОК

1. Фу К. С. Структурные методы и распознавание образов. – М.: Мир, 1977. – 318 с.
2. Ильман В. М. Структурный подход до проблеми відтворення графіків / В. М. Ильман, В. І. Шинкаренко // Проблеми програмування, 2007. – № 1. – С. 5–16.
3. Ильман В. М. Властивості формальних структур та їх підструктур // Вісник Дніпропетр. нац. ун-ту заліз. трансп. ім. акад. В. Лазаряна. – Д., 2007. – Вип. 14. – С. 99–104.
4. Ильман В. М. Некоторые приложения формальных структур / В. М. Ильман, В. В. Скалозуб, В. И. Шинкаренко // Современные информационные технологии на транспорте, в промышленности и образовании: Междунар. науч.-практ. конфер. Тезисы докладов. – Д., 2007. – С. 63–63.
5. Свами М. Графы, сети и алгоритмы / М. Свами, К. Тхуласираман – М.: Мир, – 1984 – 380 с.
6. Босов А. А. Функции множества и их применение. – Днепропетр. изд. дом «Андрей», – 2007. – 186 с.

Надійшла до редколегії 26.07.2007.