

УДК 519.7

DOI: 10.37069/1683-4720-2025-39-4

©2025. О.С. Сенченко, М.І. Притула

## ВЛАСТИВОСТІ КОМПРЕСІЇ ПАРИ, ЩО Є ВИЗНАЧАЛЬНОЮ ДЛЯ ДЕТЕРМІНОВАНОГО ГРАФА

У лінгвістичному представленні детермінованих та сильно-детермінованих графів визначальною парою множин слів запропоновано еквівалентні перетворення, які, при застосуванні їх до коректної пари, не змінюють граф, для якого ця пара є визначальною. На основі цих елементарних перетворень розроблено алгоритм компресії, який перетворює вихідну коректну пару в канонічну визначальну пару для графа, що визначає вихідна пара. На основі алгоритму компресії запропоновано розв'язок задачі характеристики коректної пари без безпосередньої побудови графа за парою. Наведено властивості алгоритму компресії. Окреслено можливі напрямки застосування компресії для розв'язання деяких прикладних задач, пов'язаних з детермінованими графами.

**MSC:** 68R10, 05C62, 05C78.

**Ключові слова:** компресія, детерміновані графи, представлення, визначальна пара.

### 1. Вступ.

Детерміновані та сильно-детерміновані графи активно застосовуються в якості моделі багатьох програмних об'єктів, інформаційних процесів, топології цифрових мереж тощо. У зарубіжних дослідженнях графи, які є подібними до детермінованих та сильно-детермінованих мають назву «graphs with injective coloring» та «graphs with distance-2 coloring» відповідно. Зокрема, сильно-детерміновані графи широко використовуються [1] для планування розміщення структурних елементів та каналів бездротових мереж. Таке планування забезпечує ефективність роботи мереж за низкою напрямків: побудови мінімального покриття, тобто визначення оптимальної кількості веж та їх розміщення, оптимізації маршрутизації, керування інтерференцією для зменшення радіоперешкод, балансування навантаження, оптимізації енергоспоживання тощо. На даний момент увага зарубіжних дослідників в основному зосереджена на оцінці оптимальності побудови дводистантної та/або ін'єктивної розмітки (оцінювання хроматичного числа, складності алгоритму тощо). Проте, в контексті практичних застосувань таких графів, важливою задачею є дослідження структури та властивостей детермінованих та сильно-детермінованих графів.

У роботі [2] розроблено лінгвістичне представлення детермінованих та сильно-детермінованих графів, які далі в роботі ми будемо називати Д-графами та СД-графами відповідно, за допомогою двох множин слів особливого вигляду, які було названо терміном «пара»: визначено алгоритм переходу від пари до графа та алгоритм переходу від графа до особливої пари, яку було названо канонічною. Таке

---

This work was supported by a grant from the Simons Foundation (PD-Ukraine-00010584, Oleksii Senchenko, Mykola Prytula)

представлення є альтернативним (відмінним від списку суміжності, матриці суміжності, списку ребер) способом задання графа з використанням лише символічних рядків, що значно скорочує розмір такого задання. Крім того, для дослідження структури об'єктів, які можна змоделювати графом з розміченими вершинами (включаючи Д- та СД-графи), часто використовують один або декілька мобільних агентів з обмеженою пам'яттю, які поміщають у досліджуваний об'єкт [3]. Розроблений у [2] спосіб представлення є досить зручним для подібного дослідження. Також у [2] окреслено ряд задач, пов'язаних з таким представленням, серед яких є задача характеристизації пари: для заданого Д- або СД-графа та заданої пари визначити, чи є ця пара визначальною для графа без безпосередньої побудови графа за парою. У цій роботі розроблено процедуру компресії, яка для коректної пари є її еквівалентним перетворенням, що призводить до спрощення пари за потужністю та об'ємом, а також визначено деякі властивості компресії. Ця процедура є розв'язком задачі характеристизації коректної пари. Крім того, на нашу думку, компресія може бути основою для розв'язання інших задач, зокрема, задачі побудови найкоротших шляхів між вершинами Д- або СД-графа.

## 2. Основні означення.

У роботі розглядаються неорієнтовані, скінченні, непорожні, зв'язні, ініціальні графи з розміченими вершинами, що не містять петлі та кратні ребра  $G = (V, E, X, \xi(V), v_0)$ , де  $V$  – множина вершин графа,  $E$  – множина його ребер,  $\xi(V) : V \rightarrow X$  – всюди визначена функція розмітки вершин графа символами скінченного алфавіту  $X = \{x_1, \dots, x_p\}$ ,  $v_0$  – ініціальна вершина. Значення функції розмітки для вершини  $v$  будемо називати її міткою. Відкритим околom вершини  $v \in V$  є множина  $E(v) = \{u | (u, v) \in E\}$ , закритим околom цієї вершини є множина  $E[v] = E(v) \cup \{v\}$ . Потужність  $|E(v)|$  називається ступенем вершини  $v$ , вершину ступеня 1 називатимемо висячою.

Граф  $G$  називається Д-графом [2], [4], якщо всі вершини у відкритому околi кожної його вершини мають попарно різні мітки:  $\forall v_1, v_2 \in E(v) (v_1 \neq v_2 \rightarrow \xi(v_1) \neq \xi(v_2))$ . Д-граф  $G$  називається СД-графом [2], [4], якщо всі вершини у замкненому околi кожної його вершини мають попарно різні мітки:  $\forall v_1, v_2 \in E[v] (v_1 \neq v_2 \rightarrow \xi(v_1) \neq \xi(v_2))$ . Інакше кажучи, у Д-графі можуть існувати дві суміжні вершини з однаковими мітками, а в СД-графі таких вершин не може бути. Зазначимо, що розмітка вершин Д-графів традиційно називається ін'єктивною розміткою, вперше така розмітка згадується у [5]; розмітка вершин СД-графів традиційно називається дводистантною розміткою, вперше така розмітка як окремий випадок  $p$ -дистантної розмітки згадується у [6]. Назви «Д-» та «СД-» граф, які є скороченими позначеннями понять «детермінований» та «сильно-детермінований» граф, пов'язані з тим, що наші дослідження передусім зосереджено на вивченні структури графа із можливим експериментальним використанням мобільних агентів, які спочатку поміщають в ініціальну вершину, і далі вони переміщуються ребрами графа та повідомляють оператору локальну інформацію (найчастіше - про мітки поточної вершин та/або мітки вершин, які їй суміжні). При цьому для Д-графів при таких переміщеннях оператор здатен однозначно визначити вершину, в якій пере-

буває агент, використовуючи лише локальну інформацію про мітку його поточної вершини, тому назва «детермінований» (тобто, однозначно визначений), на нашу думку, вдало описує такі графи. При цьому у графах, які ми досліджуємо, виокремлюється ініціальна вершина, яка відіграє ключову роль в усіх операціях; у закордонних дослідженнях жодна з вершин графа не відокремлюється, тобто поняття «ініціальна вершина» не визначається, тому поняття «Д-» та «СД-» граф не є тотожним поняттям граф з відповідно ін'єктивною та дводистантною розміткою.

Ізоморфізмом Д-графів  $G_1 = (V_1, E_1, X, \xi_1(V_1), v')$  і  $G_2 = (V_2, E_2, X, \xi_2(V_2), v'')$  з однаковою множиною міток вершин  $X$  називаємо взаємно однозначну відповідність  $\varphi : V_1 \leftrightarrow V_2$  між множинами їх вершин, що зберігає їх ініціальні вершини, відношення суміжності вершин та функцію розмітки вершин:  $\varphi(v') = v''$ ,  $\forall v_1, v_2 \in V_1 \quad ((v_1, v_2) \in E_1 \leftrightarrow (\varphi(v_1), \varphi(v_2)) \in E_2)$  та  $\forall v \in V_1 \quad \xi_1(v) = \xi_2(\varphi(v))$ ; ізоморфізм графів  $G_1$  та  $G_2$  будемо позначати  $G_1 \cong G_2$ .

У подальшому, графи, що розглядаються в цій роботі, досліджуються за допомогою слів у алфавіті їх міток. Множину всіх слів скінченної довжини в алфавіті  $X$  будемо позначати через  $X^*$ . Нехай  $p, q \in X^*$ , тоді конкатенацію слів  $p$  та  $q$  позначатимемо  $pq$ . Нехай  $p = x'_1 \dots x'_k$  ( $x'_i \in X$ ), тоді довжину слова  $p$  будемо позначати через  $d(p)$ . Зафіксуємо вершину  $v'_1$  Д-графа  $G = (V, E, X, \xi(V), v_0)$ . Оскільки шляху  $p = v'_1 \dots v'_k$  у Д-графі однозначно відповідає слово в алфавіті міток  $\xi(p) = \xi(v'_1) \xi(v'_2) \dots \xi(v'_k)$ , то надалі шлях  $p$  розглядатимемо як  $\xi(p)$  та позначатимемо рівністю  $v'_1 \xi(p) = v'_k$ . Шлях (слово)  $p = x'_1 x'_2 \dots x'_k$  назвемо припустимим для вершини  $v'_1 \in V$ , якщо  $\xi(v'_1) = x'_1$ , та існують такі вершини  $v'_2, \dots, v'_k \in V$ , що  $\xi(v'_2) = x'_2, \dots, \xi(v'_k) = x'_k$ , і  $(v'_1, v'_2), \dots, (v'_{k-1}, v'_k) \in E$ . Слово  $x'_k \dots x'_1$  будемо позначати через  $p^{-1}$ , слова  $p$  та  $p^{-1}$  назвемо взаємнозворотними. Слово  $q \in X^*$  назвемо початковим відрізком слова  $p$  у випадку, якщо існує таке слово  $w$ , що  $qw = p$ , цей факт позначатимемо через  $q \subseteq p$ . Нехай  $p = x'_1 x'_2 \dots x'_k$  – деяке слово. Якщо  $x'_i = x'_{i+2}$  ( $i + 2 \leq k$ ), то послідовність  $x'_i x'_{i+1} x'_{i+2}$  назвемо реверсом. Якщо  $p$  містить реверс та  $d(p) = 3$ , то назвемо  $p$  словом-реверсом.

Далі вважаємо, що всі слова, що розглядаються у цій роботі, розпочинаються з ініціальної вершини  $v_0$ . Нехай  $p \in X^*$  та  $d(p) = k$  та  $k > 1$ . Якщо  $k$  непарне, то через  $LEFT(p)$  позначимо номер  $\frac{k+1}{2}$  символа слова  $p$ , якщо  $k$  парне, то через  $LEFT(p)$  позначимо номер  $\frac{k}{2}$  символа слова  $p$ . Іншими словами,  $LEFT(p) = \lceil \frac{k}{2} \rceil$  порядковий номер символа слова  $p$ . Через  $RIGHT(p)$  позначимо порядковий номер символа слова  $p$ , який є в ньому наступним після символа з номером  $LEFT(p)$ :  $RIGHT(p) = \lceil \frac{k+2}{2} \rceil$ . Через  $\alpha(p)$  позначимо початковий відрізок слова  $p$  довжини  $LEFT(p)$ , через  $\beta(p)$  позначимо початковий відрізок слова  $p^{-1}$  довжини  $d(p) - LEFT(p)$ . Неважко бачити, що  $p = \alpha(p)(\beta(p))^{-1}$ .

Зафіксуємо на  $X$  лінійний порядок  $<: x_1 < x_2 < \dots < x_p$ . Визначимо на  $X^*$  лінійний порядок  $\preceq$ :

- 1) для будь-якого  $x_i \in X$ , де  $1 \leq i \leq p$ , покладемо  $x_i \preceq x_i$ ;
- 2) для будь-яких  $p, q \in X^*$ , якщо  $d(p) < d(q)$ , то  $p \preceq q$ ;
- 3) для будь-яких  $p, q \in X^*$ , якщо  $p = x'_1 \dots x'_s$ ,  $q = x''_1 \dots x''_s$ ,  $x'_1 = x''_1, \dots, x'_{k-1} = x''_{k-1}$  та  $x'_k < x''_k$ , то  $p \preceq q$ .

Також визначимо на  $X^*$  лінійний порядок  $\preceq$ :

- 1) для будь-якого  $x_i \in X$ , де  $1 \leq i \leq p$ , покладемо  $x_i \preceq x_i$ ;
- 2) для будь-яких  $p, q \in X^*$ , якщо  $d(p) < d(q)$ , то  $p \preceq q$ ;
- 3) нехай  $d(p) = d(q)$ . У випадку  $\alpha(p) \neq \alpha(q)$ , якщо  $\alpha(p) = x'_1 \dots x'_{LEFT(p)}$ ,  $\alpha(q) = x''_1 \dots x''_{LEFT(q)}$ ,  $x'_1 = x''_1, \dots, x'_{k-1} = x''_{k-1}$  та  $x'_k < x''_k$ , тоді  $p \preceq q$ . У випадку  $\alpha(p) = \alpha(q)$ , якщо  $\beta(p) = x'_1 \dots x'_s$ ,  $\beta(q) = x''_1 \dots x''_s$ ,  $x'_1 = x''_1, \dots, x'_{k-1} = x''_{k-1}$  та  $x'_k < x''_k$ , тоді  $p \preceq q$ .

Неважко бачити, що порядки  $\preceq$  та  $\preceq$  на множині  $X^*$  визначається лінійним порядком  $<$  на множині  $X$ .

Під терміном «пара»  $\{C, L\}(x')$  розуміємо дві скінченні множини слів  $C$  та  $L$ , що відповідають умовам:

- а) будь-яке слово з множини  $C$  починається та закінчується символом  $x'$ ;
- б) будь-яке слово з множини  $L$  починається символом  $x'$ ;
- в) довжина будь-якого слова з множини  $C$  більша ніж 2;
- г) довжина будь-якого слова з множини  $L$  більша ніж 1.

Множини  $C$  та  $L$  будемо називати компонентами пари  $\{C, L\}(x')$ .

У роботі [2] розроблено лінгвістичне представлення Д- та СД- графів так званою визначальною парою. На нашу думку таке представлення є досить зручним у дослідженні структури графу за допомогою мобільних агентів. Наведемо основні концепти цього представлення.

Нехай  $G = (V, E, X, \xi(V), v_0)$  – деякий Д-граф та  $\xi(v_0) = (x')$ . Представленням  $G$  є пара  $\{C, L\}(x')$ , яка задовольняє певним умовам, що стосується результату виконання алгоритму АП, який визначено у [2].

#### Початок роботи алгоритму АП.

**Вхідні дані:** пара  $\{C, L\}(x')$ .

**Вихідні дані:** Д-граф  $G(\{C, L\}(x'))$  або сповіщення, що граф  $G(\{C, L\}(x'))$  не існує.

**Крок 1.** Упорядкування слів у  $C$  та  $L$  за порядком  $\preceq$ .

**Крок 2.** Покладемо, що спочатку граф  $G(\{C, L\}(x'))$  складається з однієї вершини  $v_0$  з міткою  $\xi(v_0) = x'$ .

**Крок 3.** У відповідність кожному слову  $p^i = x'x_1^i \dots x_n^i x' \in C$  додаємо до графа вершини  $v_1^i, \dots, v_n^i$  з мітками відповідно  $x_1^i, \dots, x_n^i$  та ребра  $(v_0, v_1^i), (v_1^i, v_2^i), \dots, (v_{n-1}^i, v_n^i), (v_n^i, v_0)$ . Після кожного такого додавання робимо редукцію одержаного графа за допомогою алгоритму АР.

**Крок 4.** Для кожного слова  $p^j = x'x_1^j \dots x_n^j \in L$  виконуємо таку послідовність дій: додаємо вершини  $v_1^j, \dots, v_n^j$  з мітками відповідно  $x_1^j, \dots, x_n^j$ , ребра  $(v_0, v_1^j), \dots, (v_{n-1}^j, v_n^j)$  та робимо редукцію одержаного графа за допомогою алгоритму АР.

**Крок 5.** Розглядаємо всі слова множини  $L$ : якщо існує таке  $p \in L$ , що вершина  $v_0 p$  не є висячою або  $v_0 p = v_0$ , то граф  $G(\{C, L\}(x'))$  не існує, алгоритм АР завершує свою роботу.

**Крок 6.** Для кожної висячої вершини  $v \in G(\{C, L\}(x'))$  розглядаємо слова компоненти  $L$ : якщо не існує такого  $p \in L$ , що вершина  $v = v_0 p$ , то граф  $G(\{C, L\}(x'))$

не існує, алгоритм АР завершує свою роботу.

**Крок 7.** Для кожного слова  $p \in C \cup L$  розглядаємо припустимість шляхів  $v_0p$ : якщо шлях  $p$  не є припустимим для  $v_0$  у графі, який одержано на кроці 4, то граф  $G(\{C, L\}(x'))$  не існує, алгоритм АР завершує свою роботу.

**Кінець роботи алгоритму АП.**

Сутність редукції, алгоритм якої наведено у [2], полягає в ототожненні (редукованому злитті) вершин з однаковими мітками та ребер, що інцидентні цим вершинам, для таких вершин, що мають спільного сусіда. В процесі роботи цього алгоритму вершини впорядковуються за найкоротшим (за  $\preceq$ ) шляхом до ініціальної вершини. Далі для поточної вершини  $v_c$  у відповідності до кожної мітки  $x$  відбувається аналіз вершин з околу  $E(v_c)$ : якщо існують такі різні вершини  $v'_1, \dots, v'_q \in E(v_c)$ , що  $\xi'(v'_1) = \dots = \xi'(v'_q) = x$ , то виконується редукване злиття вершин  $v'_1, \dots, v'_q$ . Якщо в результаті виконання алгоритму АП за парою  $\{C, L\}(x')$  можливо побудувати граф  $G(\{C, L\}(x'))$ , то таку пару назвемо коректною. У алгоритмі АП крок 3 та крок 4 створюють деякий граф, який, за умови успішної перевірки на кроках 5, 6, 7 є графом  $G(\{C, L\}(x'))$ . Якщо хоча б одна з перевірок на кроках 5, 6, 7 не виконується, то граф  $G(\{C, L\}(x'))$  не існує. При цьому для скінченного графа алгоритм АП виконується скінченну кількість кроків та однозначно визначає результат. Якщо хоч одна з цих перевірок не успішна, то за даною парою представлення не існує, у цьому випадку пару  $\{C, L\}(x')$  назвемо некоректною. Коректну пару  $\{C, L\}(x')$  назвемо визначальною для Д-графа  $G$ , якщо  $G(\{C, L\}(x')) \cong G$ . Іншими словами, для того, щоб  $\{C, L\}(x')$  було представленням  $G$ , треба щоб результат виконання алгоритму АП з  $\{C, L\}(x')$  був визначеним та ізоморфним графу  $G$ .

Таким чином, представлення  $\{C, L\}(x')$  Д-графа  $G$  задовольняє вимогам:

- а) всі слова з  $C$  та  $L$  є припустимими для вершини  $v_0$ ;
- б) Кожне слово  $p$  з  $C$  визначає у  $G$  цикл  $v_0p = v_0$ ;
- в) Для кожного слова  $q \in L$  у  $G$  вершина  $v_0q$  є висячею та  $v_0q \neq v_0$ ;
- г) Для кожної висячої вершини  $v \in V$ , що є відмінною від ініціальної вершини  $v_0$ , існує хоча б одне таке слово  $q \in L$ , що  $v_0q = v$ ;
- д) для будь-якої пари  $\{C, L\}(x')$  або не існує представлення, або це представлення визначається однозначно.

При цьому різним парам може відповідати одне й те ж представлення.

У роботі [2] розроблено алгоритм (АК) побудови канонічної визначальної пари для Д-графа. Нехай  $G = (V, E, X, \xi(V), v_0)$  – деякий Д-граф, та  $V = \{v_0, \dots, v_{n-1}\}$ . Визначимо допоміжну множину слів в алфавіті  $X$ . Базисом досяжності  $\mathcal{V}_G$  назвемо таку множину слів  $\{w_1, w_2, \dots, w_n\}$ , де для кожної вершини  $v_i$  існує таке слово  $w_i \in \mathcal{V}_G$ , що виконується  $v_0w_i = v_i$ , та для будь-якого  $w \neq w_i$  з  $v_0w = v_i$ , випливає  $w_i \preceq w$ . Кістякове дерево графа  $G$ , яке визначається базисом  $\mathcal{V}_G$ , позначимо  $T(\mathcal{V}_G)$  або  $T(G, v_0)$ . Для кожної вершини  $v \in V$  через  $sp_v(G)$  позначимо таке слово з  $\mathcal{V}_G$ , що  $v_0sp_v(G) = v$ . Якщо з контексту зрозуміло, для якого графу визначається шлях  $sp_v(G)$ , то будемо позначати цей шлях  $sp_v$ .

Наведемо нотацію алгоритму АК побудови для Д-графа  $G$  пари  $\{\Sigma_G, \Lambda_G\}$ .

**Початок роботи алгоритму АК.**

**Вхідні дані:** детермінований граф  $G = (V, E, X, \xi(V), v_0)$ .

**Вихідні дані:** пара  $\{\Sigma_G, \Lambda_G\}(\xi(v_0))$ .

**Крок 1.** Спочатку покладемо  $\Sigma_G = \emptyset$  і  $\Lambda_G = \emptyset$ .

**Крок 2.** Якщо граф  $G$  складається з однієї вершини  $v_0$ , то покладемо  $\Sigma_G = \emptyset$ ,  $\Lambda_G = \emptyset$  і  $\mathcal{V}_G = \{\xi(v_0)\}$ .

**Крок 3.** Нехай граф  $G$  містить більше ніж одну вершину. Спочатку до множини  $\Lambda_G$  додаємо усі слова  $w \in \mathcal{V}_G$  такі, що вершина  $v_0 w$  є висячою вершиною графа  $G$ .

**Крок 4.** Для кожної двійки слів  $p, q \in \mathcal{V}_G \setminus \Lambda_G$ , якщо жодне з них не є початковим відрізком іншого і  $v_0 p q^{-1} = v_0$ , то додаємо до множини  $\Sigma_G$  одне з двох слів  $p q^{-1}$  або  $q p^{-1}$ , яке є меншим за порядком  $\preceq$ .

**Кінець роботи алгоритму АК.**

Нехай  $p = \alpha(p)(\beta(p))^{-1}$  – деяке слово з  $p \in X^*$  та  $G = (V, E, X, \xi(V), v_0)$  – деякий Д-граф. Якщо кожний початковий відрізок  $\alpha(p)$  та  $\beta(p)$  належить базису досяжності  $\mathcal{V}_G$  та  $p$  не містить реверсів, то назвемо  $p$  канонічним словом для  $G$ .

З цього означення випливають наступні твердження.

**Лема 1.** Нехай  $p$  є канонічним словом для графа  $G$ . Якщо  $d(p)$  є парним числом, то  $p^{-1}$  також є канонічним словом для графа  $G$ ; якщо ж  $d(p)$  є непарним числом, то  $p^{-1}$  не є канонічним словом для графа  $G$ .

*Доведення.* Нехай  $p$  – канонічне слово для графа  $G$  та  $d(p)$  – парне число. Тоді, за означеннями  $\alpha(p)$  та  $\beta(p)$ , виконуються рівності  $\alpha(p^{-1}) = \beta(p)$  та  $\beta(p^{-1}) = \alpha(p)$ , тоді  $p^{-1} = \beta(p)(\alpha(p))^{-1}$ , тому  $p^{-1}$  є канонічним словом для графа  $G$ . Нехай тепер  $d(p)$  – непарне число. Тоді, за означеннями  $\alpha(p)$  та  $\beta(p)$ ,  $v_0 \alpha(p) = v_0 \alpha(p^{-1})$ . З того, що  $p$  є канонічним словом для графа  $G$ , випливає  $\alpha(p) \in \mathcal{V}_G$ , тому  $\alpha(p^{-1}) \notin \mathcal{V}_G$ , тобто  $p^{-1}$  не є канонічним словом для графа  $G$ .  $\square$

**Лема 2.**  $\Sigma_G$  складається з усіх канонічних для графа  $G$  слів непарної довжини та таких канонічних для  $G$  слів парної довжини  $p$ , що  $p \preceq p^{-1}$ .

Відмітимо, що при побудові  $\Sigma_G$  у випадку, коли довжина одного з слів  $p$  або  $q^{-1}$  є парним числом, а довжина іншого є непарним числом та  $p q^{-1} \preceq q p^{-1}$ , то слово  $q p^{-1}$  не є канонічним, оскільки  $\alpha(q p^{-1}) = q \text{ LEFT}(q p^{-1})$  не є найкоротшим за  $\preceq$  словом для вершини  $v_0 \alpha(q p^{-1})$ , для цієї вершини таким словом є слово  $p$ .

Доведемо, що пара  $\{\Sigma_G, \Lambda_G\}$  є визначальною для графа  $G$ . Справедливе твердження.

**Теорема 1.**  $G(\{\Sigma_G, \Lambda_G\}(\xi(v_0))) \cong G$ .

*Доведення.* Для доведення теореми введемо наступні позначення. Впорядкуємо слова в  $\Sigma_G$  та  $\Lambda_G$  за порядком  $\preceq$ . Нехай  $\Sigma_G = \{\sigma_1, \dots, \sigma_p\}$ ,  $\Lambda_G = \{\lambda_1, \dots, \lambda_q\}$ . Позначимо сімейство  $G_{i,j} = (V_{i,j}, E_{i,j}, X, \xi_{i,j}(V_{i,j}), v_0)$  підграфів графа  $G$  ( $G_{0,0} \subset G_{1,0} \subset \dots \subset G_{p,0} \subset G_{p,1} \subset \dots \subset G_{p,q} \subseteq G$ ): через  $G_{0,0}$  позначимо граф, який складається з однієї вершини  $v_0$  з міткою  $\xi(v_0) = x'$ ; через  $G_{1,0}$  позначимо підграф графа  $G$ , який складається з вершин та ребер, що входять до шляху  $v_0 \sigma_1$ ; через  $G_{2,0}$  – підграф графа  $G$ , який складається з вершин та ребер, що входять до шляхів



$v_0\sigma_1$  та  $v_0\sigma_2, \dots$ ; через  $G_{p,0}$  – підграф графа  $G$ , який складається з вершин та ребер, що входять до шляхів  $v_0\sigma_1, \dots, v_0\sigma_p$ . Через  $G_{p,1}$  позначимо підграф графа  $G$ , який складається з вершин та ребер, що входять до шляхів  $v_0\sigma_1, \dots, v_0\sigma_p$  та  $v_0\lambda_1$ ; через  $G_{p,2}$  – підграф графа  $G$ , який складається з вершин та ребер, що входять до шляхів  $v_0\sigma_1, \dots, v_0\sigma_p, v_0\lambda_1$  та  $v_0\lambda_2, \dots$ ; через  $G_{p,q}$  – підграф графа  $G$ , який складається з вершин та ребер, що входять до шляхів  $v_0\sigma_1, \dots, v_0\sigma_p, v_0\lambda_1, \dots, v_0\lambda_q$ . З алгоритму АК випливає рівність  $G = G_{p,q}$ . Також введемо сімейство  $F_{i,j} = (V_{i,j}^F, E_{i,j}^F, X, \xi_{i,j}^F(V_{i,j}^F), v_0)$  графів ( $F_{0,0} \subset F_{1,0} \subset \dots \subset F_{p,0} \subset F_{p,1} \subset \dots \subset F_{p,q}$ ). Через  $F_{0,0}$  позначимо граф, який складається з однієї вершини  $v_0$  з міткою  $x'$ , через  $F_{1,0}, \dots, F_{p,0}, F_{p,1}, \dots, F_{p,q}$  – графи, які утворюються при виконанні кроків 3 та 4 алгоритму АП відповідно з словами  $\sigma_1, \dots, \sigma_p, \lambda_1, \dots, \lambda_q$  пари  $G(\{\Sigma_G, \Lambda_G\})(\xi(v_0))$ . Доведемо, що для будь-яких  $i, j$ , що задовольняють умовам  $0 \leq i \leq p, j = 0$  у випадку  $i < p$  та  $0 \leq j \leq p$  у випадку  $i = p$ , виконується  $G_{i,j} \cong F_{i,j}$ . Спочатку індукцією за  $i$  доведемо  $G_{i,0} \cong F_{i,0}$ .

Виконання бази індукції ( $G_{0,0} \cong F_{0,0}$ ) тривіальне, доведемо індуктивний крок. Припустимо, для всіх таких  $y$ , що  $0 \leq y < p$ , виконується  $G_{y,0} \cong F_{y,0}$ . Доведемо, що  $G_{y+1,0} \cong F_{y+1,0}$ . Нехай  $v_l, v_r$  – твірні вершини слова  $\sigma_{y+1}$ ,  $\xi(v_l) = x_l, \xi(v_r) = x_r, \sigma_{y+1} = w_l(w_r)^{-1}$ , де  $v_0w_l = v_l$  та  $v_0w_r = v_r$ . Позначимо  $w_l = x_0^l x_1^l \dots x_a^l$  та  $w_r = x_0^r x_1^r \dots x_b^r$  ( $x_a^l = x_l, x_b^r = x_r, x_0^l = x_0^r = x'$ ). Тоді виконується один з чотирьох взаємовиключних випадків:

- а)  $v_l, v_r \in V_{y,0}$ ;
- б)  $v_l \in V_{y,0}, v_r \notin V_{y,0}$ ;
- в)  $v_r \in V_{y,0}, v_l \notin V_{y,0}$ ;
- г)  $v_l \notin V_{y,0}, v_r \notin V_{y,0}$ .

Розглянемо випадок (а). Оскільки  $w_l x_r \notin \mathcal{V}_G$  та  $w_r x_l \notin \mathcal{V}_G$  ( $v_0w_l x_r = v_r, v_0w_r x_l = v_l$  для вершин  $v_r$  та  $v_l$  такими словами з  $\mathcal{V}_G$  є відповідно  $w_l$  та  $w_r$ ), то за алгоритмом АП та визначення сімейства  $G_{i,j}$ , ребро  $(v_l, v_r)$  не входить до жодного шляху, що визначається словами з  $\Sigma_G$ , окрім слова  $\sigma_{y+1}$ , тому  $(v_l, v_r) \notin E_{y,0}$ , та шляхи  $w_l x_r$  і  $w_r x_l$  не є припустимими для вершини  $v_0$  у графі  $G_{y,0}$ . З іншого боку, за побудовою графа  $G_{y,0}$  з  $v_l, v_r \in V_{y,0}$  та  $w_l, w_r \in \mathcal{V}_G$  випливає належність графу  $G_{y,0}$  усіх вершин та ребер, що входять до шляхів  $v_0w_l$  та  $v_0w_r$ . Іншими словами, шляхи  $w_l$  та  $w_r$  є припустимими для вершини  $v_0$  у графі  $G_{y,0}$ . Таким чином, за виконання випадку (а) граф  $G_{y+1,0}$  є результатом додавання до  $G_{y,0}$  ребра  $(v_l, v_r)$ .

Розглянемо тепер побудову графа  $F_{y+1,0}$ . За визначенням сімейства  $F_{i,j}$  граф  $F_{y+1,0}$  є результатом виконання кроку 3 алгоритму АП у відповідності до слова  $\sigma_{y+1}$  – додавання до графа  $F_{y,0}$  вершин та ребер, що входять до шляху  $v_0\sigma_{y+1}$  та редукції одержаного у такий спосіб графа. Оскільки за індуктивним припущенням  $G_{y,0} \cong F_{y,0}$ , то з урахуванням наведеного у попередньому абзаці, шляхи  $w_l$  та  $w_r$  є припустимими для вершини  $v_0$  у графі  $F_{y,0}$ , а шляхи  $w_l x_r$  і  $w_r x_l$  не є припустимими для вершини  $v_0$  у графі  $F_{y,0}$ . Результатом першої ітерації редукції є редуковане злиття або трьох вершин з мітками  $x_1^l$  у випадку  $x_1^l = x_1^r$ , або двох вершин з мітками  $x_1^l$  у випадку  $x_1^l \neq x_1^r$  та  $x_1^l < x_1^r$ , або двох вершин з мітками  $x_1^r$  у випадку  $x_1^l \neq x_1^r$  та  $x_1^r < x_1^l$ , а також редуковане злиття відповідних ребер. В останніх

двох випадках результатом наступної ітерації редукції є редуковане злиття двох вершин з мітками  $x_1^r$  та  $x_1^l$  відповідно, а також редуковане злиття відповідних ребер. У такий самий спосіб виконуються наступні ітерації редукції. Після цього у зв'язку з неприпустимістю шляхів  $w_l x_r$  та  $w_r x_l$  для вершини  $v_0$  у графі  $F_{y,0}$ , процедура редукції завершується. Отже, результатом редукції є редуковане злиття вершин та ребер, що входять до шляхів  $v_0 w_l$  та  $v_0 w_r$  з існуючими у графі  $F_{y,0}$  відповідними вершинами та ребрами. Таким чином, граф  $F_{y+1,0}$  є результатом додавання до графа  $F_{y+1,0}$  ребра  $(v_l, v_r)$ , тому виконується  $G_{y+1,0} \cong F_{y+1,0}$ , що доводить індуктивний крок для випадку (а).

Нехай тепер виконується випадок (б). Оскільки за припущенням  $w_r$  не є припустимим для вершини  $v_0$  у графі  $G_{y,0}$ , а  $x_0^r = x'$  є припустимим для цієї вершини у цьому графі, то існує такий індекс  $t$  ( $0 < t < b$ ), що у  $G_{y,0}$  для  $v_0$  шлях  $x_0^r \dots x_t^r$  є припустимим, а шлях  $x_0^r \dots x_{t+1}^r$  не є припустимим. При цьому за припущенням у цьому графі для цієї вершини шлях  $w_l$  є припустимим. Тому граф  $G_{y+1,0}$  є результатом додавання до графу  $G_{y,0}$  вершин з мітками  $x_{t+1}^r, \dots, x_b^r$  (вершина з міткою  $x_b^r$  вже була позначена через  $v_r$ ), ребер між цими вершинами, ребра між вершинами,  $v_0 x_0^r \dots x_t^r$  та  $v_0 x_0^r \dots x_{t+1}^r$ , а також ребра  $(v_r, v_l)$ . За визначенням сімейства  $F_{i,j}$  граф  $F_{y+1,0}$  є результатом виконання кроку 3 алгоритму АП у відповідності до слова  $\sigma_{y+1}$  – додавання до графа  $F_{y,0}$ , який за індуктивним припущенням є ізоморфним графу  $G_{y,0}$ , вершин та ребер, що входять до шляху  $v_0 \sigma_{y+1}$  та редукції одержаного у такий спосіб графа. Так само, як і для випадку (а) можна показати, що шлях  $w_l x_r$  не є припустимим для вершини  $v_0$  у графі  $F_{y,0}$ . Тому, з урахуванням неприпустимості шляху  $x_0^r \dots x_{t+1}^r$  для вершини  $v_0$  у графі  $F_{y,0}$ , результатом редукції є редуковане злиття вершин та ребер, що входять до шляхів  $w_l$  та  $x_0^r \dots x_t^r$  з існуючими у графі  $F_{y,0}$  відповідними вершинами та ребрами. Таким чином, граф  $F_{y+1,0}$  є результатом додавання до графа  $F_{y+1,0}$  вершин з мітками  $x_{t+1}^r, \dots, x_b^r$ , ребер між цими вершинами, ребра між вершинами, що входять до шляхів  $v_0 x_0^r \dots x_t^r$  та  $v_0 x_0^r \dots x_{t+1}^r$ , а також ребра між вершинами, що входять до шляхів  $v_0 w_r$  та  $v_0 w_l$ . Тому виконується  $G_{y+1,0} \cong F_{y+1,0}$ , що доводить індуктивний крок для випадку (б).

Виконання індуктивного кроку для випадку (в) доводиться двоїсто до випадку (б). При виконанні випадку (д) доведення подібне до доведення випадку (б).

Таким чином, доведено, що  $G_{p,0} \cong F_{p,0}$ . Далі, індукцією за  $j$ , доведемо, що для будь-якого  $j$  ( $0 \leq j \leq q$ ), виконується  $G_{p,j} \cong F_{p,j}$ . База індукції ( $G_{p,0} \cong F_{p,0}$ ) була доведена вище, доведемо індуктивний крок. Припустимо, для всіх таких  $j$ , що  $0 \leq j < q$ , виконується  $G_{p,j} \cong F_{p,j}$ . Доведемо, що  $G_{p,j+1} \cong F_{p,j+1}$ . Нехай  $\lambda_{j+1} = x'_0 x'_1 \dots x'_a$  ( $x'_0 = x'$ ). Нехай  $v$  – така вершина графа  $G$ , що  $v_0 \lambda_{j+1} = v$ . За алгоритмом АК вершина  $v$  у графі  $G$  є висячою,  $v \neq v_0$ , та для жодного слова  $p \in \Sigma_G \cup \Lambda_G$ , відмінного від слова  $\lambda_{j+1}$ , вершина  $v$  не входить до шляху  $v_0 p$ . Тому  $v \notin V_{p,j}^F$ , тобто шлях  $\lambda_{j+1} v$  не є припустимим для вершини  $v_0$  у графі  $F_{p,j}$ . З іншого боку, шлях  $x'_0 = x'$  є припустимим для вершини  $v_0$  у графі  $F_{p,j}$ , тому, аналогічно доведеного вище випадку (б), існує такий індекс  $t$  ( $0 < t < a$ ), що у  $G_{p,j}$  для  $v_0$  шлях  $x'_0 \dots x'_t$  є припустимим, а шлях  $x'_0 \dots x'_{t+1}$  не є припустимим. Тому граф  $G_{p,j+1}$  є результатом додавання до графу  $G_{p,j}$  вершин з мітками  $x'_{t+1}, \dots, x'_a$ , ребер



між цими вершинами та ребра між вершинами, що входять до шляхів  $v_0x'_0 \dots x'_t$  та  $v_0x'_0 \dots x'_{t+1}$ .

За визначенням сімейства  $F_{i,j}$ , граф  $F_{p,j+1}$  є результатом виконання кроку 4 алгоритму АП у відповідності до слова  $\lambda_{y+1}$  – додавання до графа  $F_{p,j}$ , який за індуктивним припущенням є ізоморфним графу  $G_{p,j}$ , вершин та ребер, що входять до шляху  $v_0\lambda_{y+1}$  та редукції одержаного у такий спосіб графа. Аналогічно доведеному вище випадку (б), результатом редукції є редуковане злиття вершин та ребер, що входять до шляху  $x'_0 \dots x'_t$  з існуючими у графі  $F_{p,j}$  відповідними вершинами та ребрами. Таким чином, граф  $F_{p,j+1}$  є результатом додавання до графа  $F_{p,j}$  вершин з мітками  $x'_{t+1}, \dots, x'_a$ , ребер між цими вершинами, ребра між вершинами, що входять до шляхів  $v_0x'_0 \dots x'_t$  та  $v_0x'_0 \dots x'_{t+1}$ . Тому виконується  $G_{p,j+1} \cong F_{p,j+1}$ , що доводить індуктивний крок.

Одержали, що граф  $F_{p,q}$ , який було побудовано на кроці 4 алгоритму АП за парою  $\{\Sigma_G, \Lambda_G\}(\xi(v_0))$ , є ізоморфним графу  $G$ . З алгоритму АК випливає, що для графа  $G$  перевірки за кроками 5, 6, 7 алгоритму АП успішно виконуються: для кожного слова  $\lambda \in \Lambda_G$ , вершина  $v_0\lambda$  у графі  $G$  є висячою та  $v_0\lambda \neq v_0$ ; для кожної висячої вершини  $v$ , відмінної від  $v_0$ , у графі  $G$  існує (причому єдине) таке слово  $\lambda \in \Lambda_G$ , що  $v_0\lambda = v$ ; для кожного слова  $p \in \Sigma_G \cup \Lambda_G$  шлях  $v_0p$  є припустимим для  $v_0$  у графі  $G$ . Тому всі ці перевірки успішно виконуються для графа  $F_{p,q}$ , з чого випливає, що граф  $G(\{\Sigma_G, \Lambda_G\}(\xi(v_0)))$  існує та  $G(\{\Sigma_G, \Lambda_G\}(\xi(v_0))) \cong G$ .  $\square$

### 3. Еквівалентні перетворення коректної пари.

Коректні пари  $\{C, L\}(x')$  та  $\{C_1, L_1\}(x')$  назвемо еквівалентними, якщо  $G(\{C, L\}(x')) \cong G(\{C_1, L_1\}(x'))$ . Еквівалентність пар  $\{C, L\}(x')$  та  $\{C_1, L_1\}(x')$  позначатимемо  $\{C, L\}(x') \sim \{C_1, L_1\}(x')$ . Розглянемо деякі елементарні перетворення коректної пари, які призводять до пари еквівалентній їй, при цьому результуюча пара у деякому сенсі є спрощеною відносно вихідної пари – в результуючій парі або зменшується кількість слів, або довжина слів, або деякі слова замінюються словами, меншими за порядком  $\preceq$  чи  $\lesssim$ . Ці перетворення становитимуть основу кроків алгоритму компресії, який буде наведено в наступному розділі цієї статті. Далі вихідну коректну пару позначаємо  $\{C, L\}(x')$ .

1) Вилучення повторів слів у кожній компоненті пари. Нехай пара  $\{C_1, L_1\}(x')$  утворюється з пари  $\{C, L\}(x')$  шляхом вилучення повторів слів у першій або другій компоненті  $\{C, L\}(x')$  (з коректності  $\{C, L\}(x')$  впливає неможливість існування будь-якого слова в обох її компонентах одночасно). Нехай для визначеності,  $p$  повторюється декілька разів. Тоді результат дії алгоритму АП за повторною обробкою слова  $p$  не буде відрізнятися від аналогічного результату за словом, що було перед цим оброблено, оскільки відбудеться редуковане злиття усіх вершин та ребер, які було додано до вже побудованого фрагменту графа за повторною обробкою слова  $p$  з відповідними вершинами та ребрами, які було додано до графа за першою обробкою слова  $p$  або результатом їх редукованого злиття. Тому,  $\{C, L\}(x') \sim \{C_1, L_1\}(x')$ , при цьому пара  $\{C_1, L_1\}(x')$  містить менше слів, ніж пара  $\{C, L\}(x')$ .

2) Вилучення з першої компоненти слів-реверсів. Нехай  $p = x'yx'$ ,  $p \in C$  та

$C_1$  є результатом вилучення  $p$  з  $C$ . Покажемо, що  $\{C, L\}(x') \sim \{C_1, L\}(x')$ . При виконанні першого кроку алгоритму АП за словом  $p$  до вже побудованого фрагменту графа додається вершина з міткою  $y$  (далі цю вершину називатимемо  $v$ ) та ребро, що з'єднує цю вершину з ініціальною вершиною (ребро  $(v_0, v)$ ), після цього відбувається редукція одержаного графа. Покажемо, що в одній з компонент пари  $\{C, L\}(x')$  повинно існувати слово  $w \neq x'yx'$ , за яким перший або другий крок алгоритму АП будує ребро  $(v_0, v)$ . Від супротивного, припустимо, що такого слова не існує. У такому випадку вершина  $v$  повинна бути висячою, оскільки не може відбутися редуковане злиття жодної вершини з вершиною  $v$ . При цьому з коректності пари  $\{C, L\}(x')$  випливає існування у  $L$  такого слова  $w'$ , що  $v_0w' = v$ , при цьому останнім символом слова  $w' \in y$ , а передостаннім – символ  $x'$ . Тоді, зважаючи на те, що висяча вершина  $v$  суміжна лише вершині  $v_0$ , при обробці алгоритмом АП слова  $w'$  повинна бути побудована вершина з міткою  $y$  та ребро між цією вершиною та ініціальною вершиною  $v_0$ ; при проведенні редукції буде виконано редуковане злиття цієї вершини з вершиною  $v$  та ребра між цією вершиною та  $v_0$  – з ребром  $(v_0, v)$ ; це суперечить припущенню про неіснування слова  $w$ . Отже, в одній з компонент пари  $\{C, L\}(x')$ , а тому й у одній з компонент пари  $\{C_1, L\}(x')$  існує слово, за яким будується ребро  $(v_0, v)$ . Результат обробки всіх інших слів пари  $\{C, L\}(x')$  є ідентичними результату обробки відповідних слів пари  $\{C_1, L\}(x')$ , тому граф  $G(\{C_1, L\}(x'))$  існує та  $G(\{C, L\}(x')) \cong G(\{C_1, L\}(x'))$ , при цьому пара  $\{C_1, L\}(x')$  містить менше слів, ніж пара  $\{C, L\}(x')$ . Зауважимо, що з коректності  $\{C, L\}(x')$  випливає неможливість існування слів-реверсів у її другій компоненті.

3) Вилучення реверсів зі слів з обох компонент. Нехай слово  $p$  належить одній з компонент пари  $\{C, L\}(x')$ ,  $d(p) > 3$ , та  $p$  містить реверс  $xux$ . Покладемо  $p \in C$ . Позначимо  $p = w_1xuxw_2$ , де  $w_1, w_2 \in X^*$ ;  $p' = w_1xw_2$  – слово, яке отримано з слова  $p$  шляхом заміни реверсу його першим символом;  $C_1$  – результат заміни у  $C$  слова  $p$  словом  $p'$ . ‘Процес виконання алгоритму АП із парою  $\{C, L\}(x')$  відрізняється від такого процесу з парою  $\{C_1, L\}(x')$  обробкою слів  $p$  та  $p'$ . При виконанні першого кроку АП за словом  $p$  додається ряд вершин та ребер, що відповідають символам слова  $p$ , у тому числі вершина з міткою  $y$ , яка суміжна двом вершинам з мітками  $x$  у відповідності до реверсу  $xux$ , далі в результаті редукції відбувається редуковане злиття цих двох вершин з мітками  $x$ . При виконанні першого кроку АП за словом  $p'$  відбуваються ті ж самі дії, що й при виконанні першого кроку АП за словом  $p$ , за виключенням побудови вершини з міткою  $y$  у відповідності до реверсу  $xux$ . Таким чином, результат виконання першого кроку АП за словом  $p$  відрізняється від аналогічного результату за словом  $p'$  побудовою однієї вершини та одного ребра, позначимо їх відповідно  $v$  та  $(v, v')$ . Так само, як і для попереднього перетворення (для нашого випадку вершина  $v'$  відіграє роль, яку в попередньому перетворенні відігравала вершина  $v_0$ ), можна показати наявність в одній з компонент пари  $\{C, L\}(x')$  слова  $w \neq p$ , за яким перший або другий крок алгоритму АП будує ребро  $(v, v')$ , з чого випливає існування графа  $G\{C_1, L\}(x')$  та  $G(\{C, L\}(x')) \cong G(\{C_1, L\}(x'))$ , тобто  $\{C, L\}(x') \sim \{C_1, L\}(x')$ . У випадку  $p \in L$  та аналогічними позначеннями  $p'$  та  $L_1$  виконується еквівалентність

$\{C, L\}(x') \sim \{C, L_1\}(x')$ ; пояснення такі ж самі, як і для випадку  $p \in C$ . В результаті виконання цього перетворення результуюча пара відрізняється від вихідної одним словом, довжина цього слова в результуючій парі на 2 менше, ніж довжина слова, яке було перетворено.

4) Заміна у першій компоненті слова  $p$  словом  $p^{-1}$  у випадку  $p^{-1} \preceq p$ . Нехай  $p \in C$  та  $p^{-1} \preceq p$ . Позначимо  $C_1$  – результат заміни у  $C$  слова  $p$  словом  $p^{-1}$ . Процес виконання алгоритму АП із парою  $\{C, L\}(x')$  відрізняється від такого процесу з парою  $\{C_1, L\}(x')$  обробкою слів  $p$  та  $p^{-1}$ . Неважко бачити, що при обробці слова  $p$  до вже побудованого фрагменту графа будуть додані такі ж самі вершини та ребра, що й при обробці слова  $p^{-1}$ , тому  $\{C, L\}(x') \sim \{C_1, L\}(x')$ . В результаті виконання цього перетворення результуюча пара відрізняється від вихідної одним словом, це слово в результуючій парі менше за порядком  $\preceq$ , ніж слово, яке було перетворено. Зауважимо, що це перетворення не можна застосовувати для слів з другої компоненти пари.

5) Оптимізація шляхів у відповідності до вершини розгалуження. Нехай  $p \in C$  та не містить реверсів. На шляху  $v_0p$  зафіксуємо довільну вершину з міткою  $x$ , яку далі називатимемо вершиною розгалуження. Позначимо  $p = p_1xp_2$ , де мітка  $x$  відповідає вершині розгалуження. Тоді відносно шляху від ініціальної вершини до вершини розгалуження можливий один з двох взаємовиключних випадків щодо порівняння слів  $p_1$  та  $(p_2)^{-1}$  за порядком  $\preceq$ :

1)  $p_1 \preceq (p_2)^{-1}$ .

а) Нехай  $q \neq p$ ,  $q \in C \cup L$  та  $(p_2)^{-1}x$  є початковим відрізком слова  $q$ , тобто  $q = (p_2)^{-1}xq'$  для деякого  $q' \in X^*$ . Позначимо  $q_1 = p_1xq'$ . Через  $\{C_1, L_1\}(x')$  позначимо пару, у якій слово  $q$  замінено на слово  $q_1$ , а всі інші слова, відмінні від  $q$ , є такими ж, як і у парі  $\{C, L\}(x')$ .

б) Нехай  $q \neq p$ ,  $q \in C$  та  $q$  закінчується на  $xp_2$ , тобто  $q = q'xp_2$  для деякого  $q' \in X^*$ . Позначимо  $q_2 = q'x(p_1)^{-1}$ . Через  $\{C_2, L\}(x')$  позначимо пару, у якій слово  $q$  замінено на слово  $q_2$ , а всі інші слова, відмінні від  $q$ , є такими ж, як і у парі  $\{C, L\}(x')$ .

2)  $(p_2)^{-1} \preceq p_1$ .

а) Нехай  $q \neq p$ ,  $q \in C \cup L$  та  $p_1x$  є початковим відрізком слова  $q$ , тобто  $q = p_1xq'$  для деякого  $q' \in X^*$ . Позначимо  $q_3 = (p_2)^{-1}xq'$ . Через  $\{C_3, L_3\}(x')$  позначимо пару, у якій слово  $q$  замінено на слово  $q_3$ , а всі інші слова, відмінні від  $q$ , є такими ж, як і у парі  $\{C, L\}(x')$ .

б) Нехай  $q \neq p$ ,  $q \in C$  та  $q$  закінчується на  $x(p_1)^{-1}$ , тобто  $q = q'x(p_1)^{-1}$  для деякого  $q' \in X^*$ . Позначимо  $q_4 = q'xp_2$ . Через  $\{C_4, L\}(x')$  позначимо пару, у якій слово  $q$  замінено на слово  $q_4$ , а всі інші слова, відмінні від  $q$ , є такими ж, як і у парі  $\{C, L\}(x')$ .

Тоді, відповідно до кожного з розглянутих випадків, виконується або  $\{C, L\}(x') \sim \{C_1, L_1\}(x')$ , або  $\{C, L\}(x') \sim \{C_2, L\}(x')$ , або  $\{C, L\}(x') \sim \{C_3, L_3\}(x')$ , або  $\{C, L\}(x') \sim \{C_4, L\}(x')$ . Для цих випадків результуюча пара відрізняється від вихідної одним словом, це слово в результуючій парі у першому та третьому випадках менше за порядком  $\preceq$ , ніж слово  $q$ , яке було перетворено, а у другому та

четвертому випадках це слово менше за порядком  $\preceq$ , ніж слово  $q$ .

Наведемо пояснення щодо першої еквівалентності. Процес виконання алгоритму АП із парою  $\{C, L\}(x')$  відрізняється від такого процесу з парою  $\{C_1, L_1\}(x')$  обробкою слів  $q$  та  $q_1$ . При обробці слова  $q$  на третьому кроці алгоритму АП додаються вершини та ребра, що відповідають символам слова  $(p_2)^{-1}$ , вершини та ребра, що відповідають символам слова  $q'$ , а також вершина розгалуження, ребро між цією вершиною та вершиною  $v_0(p_2)^{-1}$  і ребро між вершиною розгалуження та вершиною  $v_0q'$ . При обробці слова  $q_1$  на третьому кроці алгоритму АП додаються вершини та ребра, що відповідають символам слова  $p_1$ , вершини та ребра, що відповідають символам слова  $q'$ , а також вершина розгалуження, ребро між цією вершиною та вершиною  $v_0p_1$  і ребро між вершиною розгалуження та вершиною  $v_0q'$ . Далі, в результаті редукції одержаного графа, у першому випадку відбувається редуковане злиття вершин та ребер, що відповідають символам слова  $(p_2)^{-1}$ , а також вершини розгалуження з відповідними вершинами, які було додано при обробці слова  $p$ . У другому випадку в результаті редукції відбувається редуковане злиття вершин що відповідають символам слова  $p_1$ , а також вершини розгалуження з відповідними вершинами, які було додано при обробці слова  $p$ . Всі інші додавання вершин та ребер, а також подальша редукція при обробці слів  $q$  та  $q_1$  є ідентичними, тому результати обробки алгоритмом АП слів  $q$  та  $q_1$  співпадають. Інші 3 еквівалентності пояснюються аналогічно.

Сформулюємо одержані результати у вигляді наступного твердження.

**Теорема 3.** *Нехай  $\{C, L\}(x')$  – коректна пара та  $\{C_1, L_1\}(x')$  одержана з  $\{C, L\}(x')$  одноразовим застосуванням будь-якого з елементарних перетворень (1) – (5). Тоді  $\{C, L\}(x') \sim \{C_1, L_1\}(x')$ .*

Усі ці елементарні перетворення є ключовими для алгоритму компресії, який наведено в наступному розділі цієї статті.

#### 4. Алгоритм компресії та його властивості.

Даний розділ роботи присвячений розв'язанню задачі характеристики коректної пари: треба визначити, чи є задана пара  $\{C, L\}(x')$  визначальною для заданого Д-графа  $G$  без безпосередньої побудови графа  $G(\{C, L\}(x'))$  та подальшого його порівняння з  $G$ . Для розв'язання цієї задачі розроблено алгоритм компресії, який в процесі своєї роботи в якості операцій використовує лише запропоновані у попередньому розділі статті елементарні перетворення, а очікуваним результатом роботи алгоритма є канонічна пара графа  $G(\{C, L\}(x'))$ . При цьому алгоритм компресії у певному сенсі подібний до алгоритму, який є розв'язком задачі характеристики системи визначальних співвідношень для скінченного автомату без виходу [7].

Зважаючи на те, що елементарні перетворення (1) – (4) в алгоритмічному сенсі є простішими, ніж перетворення (5), при розробці алгоритму компресії головну увагу було зосереджено на оптимізації перетворення (5). Покажемо, що з урахуванням виконання перетворень (1) – (4) на інших кроках алгоритму компресії, у перетворенні (5) в якості вершини розгалуження для слова  $p = x'_1x'_2 \dots x'_{k-1}x'_k \in C$  ( $x'_1 = x'_k = x'$ ) достатньо взяти вершини з індексами  $LEFT(p)$  та  $RIGHT(p)$ .

Нехай виконується випадок (1a) при елементарному перетворенні 5, тобто  $p \in C$ ,  $p = p_1xp_2$ , де мітка  $x$  відповідає вершині розгалуження,  $p_1 \preceq (p_2)^{-1}$ ,  $q \neq p$  та  $q = (p_2)^{-1}xq'$  для деякого  $q' \in X^*$ . Через  $i$  позначимо порядковий номер (індекс) символу, що відповідає вершині розгалуження у слові  $p$ . З  $p_1 \preceq (p_2)^{-1}$  та попереднього виконання елементарного перетворення 4 випливає  $1 \leq i \leq LEFT(p)$ . В результаті виконання перетворення 5 одержимо заміну слова  $q$  на слово  $r = p_1xq'$ .

Покажемо, що у випадку  $i \neq LEFT(p)$  в результаті виконання елементарного перетворення 5 за вершиною розгалуження, яка має індекс  $LEFT(p)$  та певної послідовності інших елементарних перетворень ми одержимо такий самий результат перетворення (заміни) слова  $q$ . Введемо наступні позначення. Через  $y$  позначимо фрагмент слова  $p$  від символу, що є наступним після вершини розгалуження у попередньому випадку до символу з індексом  $LEFT(p)$  (цей символ позначимо  $p(LEFT(p))$ ), не включаючи сам символ з індексом  $LEFT(p)$ , через  $w$  позначимо фрагмент слова  $p$  від символу з індексом  $LEFT(p) + 1 = RIGHT(p)$  до кінця слова  $p$ , тобто  $p = p_1xyp(LEFT(p))w$ . При цьому в цих позначеннях  $q = w^{-1}p(LEFT(p))y^{-1}xq'$ . Відбувається заміна у слові  $q$  послідовності  $w^{-1}$  на  $p_1xyp(LEFT(p))$ , тобто слово  $q$  замінюється словом  $q_1 = p_1xyp(LEFT(p))y^{-1}xq'$ . Далі, після вилучення реверсів зі слова  $q_1$  у кількості, що дорівнює довжині слова  $y$  одержимо перетворення слова  $q_1$  на слово  $q_2 = p_1xzxq'$ , де  $z$  є першим символом слова  $y$ ; у випадку, коли  $y$  є порожнім словом, то  $z = p(LEFT(p))$ , після чого відбувається вилучення у  $q_2$  реверсу  $xzx$ , тобто одержимо перетворення слова  $q_2$  на слово  $q_3 = p_1xq'$ , тобто  $q_3 = r$ .

Таким чином показано, що за умови виконання елементарних перетворень 3 та 4 (перетворення 3 виконується як до виконання перетворення 5, так і після нього), для перетворення 5 у випадку (1a) за вершину розгалуження у слові  $p$  можна взяти вершину з індексом  $LEFT(p)$ . Через  $OPT-LB(p, q)$  (скорочення від «Оптимізація шляхів у відповідності до вершини-розгалужувача, індекс якої у слові  $p$  є  $LEFT(p)$  та заміна у слові  $q$  виконується з початку цього слова») позначимо перетворення слова  $q$  за вершиною розгалуження слова  $p$ , що відповідає індексу  $LEFT(p)$ ; у нашому випадку  $OPT-LB(p, q) = p_1xyp(LEFT(p))y^{-1}xq'$ . В обчислювальному сенсі недоліком такого перетворення є додаткове, можливо, кількаразове використання перетворення 3 з одним словом – результатом  $OPT-LB(p, q)$ . Перевагою є те, що вершина розгалуження визначається однозначно, тобто для її визначення не треба використовувати цикл від першого символу слова  $p$  до символу з індексом  $LEFT(p)$  з подальшим розгляданням усіх слів з компонент  $C$  та  $L$ .

Для випадку (1б) результат перетворення у відповідності до вершини розгалуження з індексом  $i$  у слові  $p$  ( $1 \leq i < LEFT(p)$ ) буде дорівнювати такому ж перетворенню у відповідності до вершини розгалуження з індексом  $LEFT(p)$  з подальшим, можливо кількаразовим, виконанням перетворення 3 (це доводиться двоїсто до випадку 1(а)). Позначимо перетворення у відповідності до вершини розгалуження з індексом  $LEFT(p)$  через  $OPT-LE(p, q)$  (скорочення аналогічне до  $OPT-LB(p, q)$ , проте заміна у слові  $q$  виконується з кінця цього слова).

Нехай виконується випадок (2a) при елементарному перетворенні 5, тобто  $p \in$

$C$ ,  $p = p_1xp_2$ , де мітка  $x$  відповідає вершині розгалуження,  $(p_2)^{-1} \preceq p_1$ ,  $q \neq p$  та  $q = p_1xq'$  для деякого  $q' \in X^*$ . Через  $i$  позначимо порядковий номер (індекс) символу, що відповідає вершині розгалуження у слові  $p$ . З  $(p_2)^{-1} \preceq p_1$  та попереднього виконання елементарного перетворення 4 випливає  $RIGHT(p) \leq i \leq d(p)$ . Аналогічно до випадку (1a), результат виконання перетворення 5 зі словом  $q$  у випадку  $i \neq RIGHT(p)$  буде такий самий як результат його виконання у випадку  $i = RIGHT(p)$  з подальшим, можливо кількарарозовим, застосуванням до нього перетворення 3. Перетворення у відповідності до вершини розгалуження з індексом  $RIGHT(p)$  позначимо  $OPT - RB(p, q)$ . Так саме, для випадку (2б) результат виконання перетворення 5 зі словом  $q$  у випадку  $i \neq RIGHT(p)$  буде такий самий як результат його виконання у випадку  $i = RIGHT(p)$  з подальшим, можливо кількарарозовим, застосуванням до нього перетворення 3; перетворення у відповідності до вершини розгалуження з індексом  $RIGHT(p)$  позначимо  $OPT - RE(p, q)$ .

З елементарних перетворень (1) – (4), які ми позначили відповідно  $DEL - REP(w)$ ,  $DEL - RW(w)$ ,  $REM - R(w)$ ,  $ORD(w)$ , сортування слів у компонентах пари  $\{C, L\}(x')$ , яке ми позначили  $SORT(\{C, L\})$  та перетворень  $OPT - LB(w, q)$ ,  $OPT - LE(w, q)$ ,  $OPT - RB(w, q)$ ,  $OPT - RE(w, q)$ , було створено алгоритм компресії. З останніх чотирьох перетворень створено складену операцію  $PATH - OPT(\{C, L\})$ . На рис. 1 наведено нотацію цієї операції.

```

DEFINE PATH-OPT({C, L}):
  FOR w INC:
    FOR q IN C:
      IF q ≠ w:
        OPT-LB(w, q)
        OPT-RB(w, q)
        OPT-LE(w, q)
        OPT-RE(w, q)
    FOR q IN L: OPT-LB(w, q) OPT-RB(w, q)

```

Рис. 1. Нотація операції  $PATH - OPT(\{C, L\})$

Визначимо алгоритм компресії, який перетворює коректну пару  $\{C, L\}(x')$  у пару  $< \{C, L\}(x') >$ .

**Початок роботи алгоритму компресії.**

**Вхідні дані:** пара  $\{C, L\}(x')$ .

**Вихідні дані:** пара  $< \{C, L\}(x') >$ .

**Крок 1.** Виконати  $SORT(\{C, L\})$  та  $DEL - REP(w)$ .

**Крок 2.** Виконати  $DEL - RW(w)$  для слів з першої компоненти пари.

**Крок 3.** Виконати  $REM - R(w)$  для слів з кожної компоненти пари. Якщо відбулось таке перетворення, то перейти на крок 1, інакше виконати наступний крок алгоритму.

**Крок 4.** Виконати  $ORD(w)$  для слів з першої компоненти пари. Якщо відбулось таке перетворення, то перейти на крок 1, інакше виконати наступний крок алгоритму.



**Крок 5.** Виконати  $PATH - OPT(\{C, L\})$ . Якщо відбулось таке перетворення, то перейти на крок 1, інакше виконати наступний крок алгоритму.

**Кінець роботи алгоритму компресії.**

Оскільки кожна з операцій алгоритму компресії, крім сортування, або зменшує кількість слів у одній з компонент пари, або зменшує довжину деякого слова, або замінює деяке слово на слово, яке є меншим його за порядком  $\preceq$  чи  $\succsim$ , то для скінченної вихідної пари алгоритм компресії за скінченну кількість кроків завершує свою роботу з однозначно визначеним результатом. Компоненти результату компресії пари  $\{C, L\}(x')$  позначимо відповідно через  $\langle C \rangle$  та  $\langle L \rangle$ .

На рис. 2 наведемо приклад роботи алгоритму компресії.

$C = \{0120, 03530, 03524150, 03114251130, 012035241130\}$

$L = \{035230, 03114230, 0351130123\}$

01. Operation REM-R(w) is performed; w = 03530, modified = 030
02. Operation DEL-RW(w) is performed; w = 030
03. Operation OPT-RB(p, q) is performed; p = 0120, q = 012035241130, modified = 02035241130
04. Operation REM-R(w) is performed; w = 02035241130, modified = 035241130
05. Operation ORD(w) is performed; w = 035241130, modified = 031142530
06. Operation OPT-RE(p, q) is performed; p = 03524150, q = 031142530, modified = 03114150
07. Operation REM-R(w) is performed; w = 03114150, modified = 031150
08. Operation OPT-RB(p, q) is performed; p = 031150, q = 03114251130, modified = 0514251130
09. Operation ORD(w) is performed; w = 0514251130, modified = 0311524150
10. Operation OPT-RB(p, q) is performed; p = 031150, q = 0311524150, modified = 051524150
11. Operation REM-R(w) is performed; w = 051524150, modified = 0524150
12. Operation ORD(w) is performed; w = 0524150, modified = 0514250
13. Operation OPT-RB(p, q) is performed; p = 031150, q = 03114230, modified = 0514230
14. Operation OPT-RE(p, q) is performed; p = 0514250, q = 03524150, modified = 035250
15. Operation REM-R(w) is performed; w = 035250, modified = 0350
16. Operation OPT-RB(p, q) is performed; p = 0350, q = 035230, modified = 05230
17. Operation OPT-RB(p, q) is performed; p = 0350, q = 0351130123, modified = 051130123
18. Operation OPT-LB(p, q) is performed; p = 031150, q = 051130123, modified = 03130123
19. Operation REM-R(w) is performed; w = 03130123, modified = 030123
20. Operation REM-R(w) is performed; w = 030123, modified = 0123
21. Operation OPT-RB(p, q) is performed; p = 0120, q = 0123, modified = 023
22. Operation OPT-RB(p, q) is performed; p = 0514250, q = 0514230, modified = 05230
23. Operation DEL-REP(w) is performed; w = 05230

$\langle C \rangle = \{0120, 0350, 031150, 0514250\}; \langle L \rangle = \{023, 05230\}$

**Рис. 2.** Приклад роботи алгоритму компресії

Сформулюємо критерій розв'язання задачі характеристики коректної пари, яку визначено на початку цього розділу роботи. Справедливе наступне твердження.

**Теорема 4.** Нехай  $\{C, L\}(x')$  – коректна пара. Тоді  $\{C, L\}(x')$  є визначальною парою для  $G$  тоді та лише тоді, коли  $\langle \{C, L\}(x') \rangle = \{\Sigma_G, \Lambda_G\}$ .

*Доведення.* Доведемо достатність. Нехай  $\{C, L\}(x')$  – коректна пара та  $\langle \{C, L\}(x') \rangle = \{\Sigma_G, \Lambda_G\}$  для деякого графа  $G$ . Позначимо процес виконання компресії пари  $\{C, L\}(x')$  як ланцюг перетворень:  $\{C, L\}(x') = \{C_0, L_0\}(x') \rightarrow \{C_1, L_1\}(x') \rightarrow \dots \rightarrow \{C_{n-1}, L_{n-1}\}(x') \rightarrow \{C_n, L_n\}(x') = \{\Sigma_G, \Lambda_G\}$ . У цьому ланц-

юзі пара  $\{C_{k+1}, L_{k+1}\}(x')$  одержана з пари  $\{C_k, L_k\}(x')$  однократним виконанням деякої операції алгоритму компресії ( $k = \overline{0, \dots, n-1}$ ). За теоремою 3 усі операції в алгоритмі компресії перетворюють деяку пару в пару, що є еквівалентною їй, (перетворення  $OPT-LB(w, q)$ ,  $OPT-LE(w, q)$ ,  $OPT-RB(w, q)$ ,  $OPT-RE(w, q)$ , які є частинами складеної операції  $PATH - OPT(\{C, L\})$ , є окремими випадками елементарного перетворення 5), а операція  $SORT(\{C, L\})$  не змінює граф  $G(\{C, L\}(x'))$ , то  $G = G(\{C_n, L_n\}(x')) = \dots = G(\{C_0, L_0\}(x'))$ , тобто  $\{C, L\}(x')$  є визначальною парою для  $G$ , що доводить достатність.

Доведемо необхідність. Нехай  $\{C, L\}(x')$  є визначальною парою для  $G$  та  $v_0$  – ініціальна вершина графа  $G$ . Доведемо допоміжне твердження

**Лема 3.**  $U < \{C, L\}(x') >$  для будь-яких слів  $p \in < C >$ ,  $q \in < L >$  будь-який початковий відрізок слів  $\alpha(p)$ ,  $\beta(p)$  та  $q$  є найкоротшим за  $\preceq$  шляхом до відповідної вершини у графі  $G(\{C, L\}(x'))$ .

*Доведення.* Доведення проведемо індукцією за довжиною  $i$  найкоротших шляхів  $sp_v(G)$  до вершин  $v$  графа  $G$ . Нехай  $i = 1$ . Оскільки для будь-якого початкового відрізка  $w$  довжини 1 слів  $\alpha(p)$ ,  $\beta(p)$ ,  $q$  ( $p \in < C >$ ,  $q \in < L >$ )  $< \{C, L\}(x') >$  виконується  $w = x'$  та  $v_0x' = v_0$ , то  $sp_{v_0w}(G) = w$ , тобто база індукції виконується.

Припустимо, що лема виконується для  $i \leq k$ . Покажемо, що вона також виконується для  $i = k + 1$ . Від супротивного, припустимо, що це не так, тобто існує таке слово  $w$ , що  $d(w) = k + 1$ ,  $w$  не є найкоротшим за  $\preceq$  словом до вершини  $v_0w$ , та виконується один з трьох випадків: а) існує таке  $p \in < C >$ , що  $w$  є початковим відрізком  $\alpha(p)$ ; б) існує таке  $p \in < C >$ , що  $w$  є початковим відрізком  $\beta(p)$ ; в) існує таке  $q \in < L >$ , що  $w$  є початковим відрізком  $q$ . Нехай виконується випадок (а).

Введемо такі позначення:  $p = p_1wp'$ ,  $v_0w = v$ ,  $\xi(v) = x$ ,  $w = p_1x$ ,  $v_0w = v'$ . Тоді  $d(p_1) = k$ , за припущенням індукції  $p_1 = sp_{v'}(G)$ . Нехай  $z = z'x = sp_v(G)$ , тоді, з урахуванням припущення індукції,  $d(z'x) = k + 1$ , тому  $d(z') = k$ ; позначимо  $v_0z' = v''$ . З  $z = sp_v(G)$  випливає  $z' = sp_{v''}(G)$  та  $z' \preceq p_1$ . З  $p_1x \neq z'x$  випливає  $p_1 \neq z'$ , тому, з урахуванням  $p_1 = sp_{v'}$ , та  $z' = sp_{v''}$ , а також припущення індукції, в графі  $G$   $v' \neq v''$ . Оскільки  $< \{C, L\}(x') >$  є визначальною парою графа  $G$ , то існує таке слово  $u$ , що  $v_0u = v''$  та для деякого  $u' \in X^*$  виконується хоча б одне з тверджень  $uxu' \in < C >$  або  $u'xu^{-1} \in < C >$ . З припущення індукції ( $d(z') = k$ ) випливає  $u = z'$ .

Тоді, з того факту, що  $< C >$  та  $< L >$  є результатом компресії пари  $\{C, L\}(x')$ , у графі  $G$  вершина, що відповідає шляху  $v_0w$  була ототожнена з вершиною, що відповідає шляху  $v_0z$  тільки за умови існування в  $< C >$  одного з слів  $z'x(p_1)^{-1}$  або  $p_1x(z')^{-1}$ . Оскільки ці слова є взаємозворотними, та  $z'x(p_1)^{-1} \preceq p_1x(z')^{-1}$ , то з виконання елементарного перетворення  $ORD$  у  $< C >$  випливає  $p_1x(z')^{-1} \notin < C >$ , тому  $z'x(p_1)^{-1} \in < C >$ . При цьому  $z'x(p_1)^{-1} \neq p$ . Тому, з виконання перетворення  $OPT-LB(z'x(p_1)^{-1}, p)$  у  $< C >$  слово  $p$  повинно було бути заміненим словом  $z'xp'$ , тобто  $p \notin < C >$ . Отримана суперечність доводить хибність припущення про невиконуваність леми для  $i = k + 1$  за умови виконання випадку (а).

Хибність припущення про невиконуваність леми для випадку (б) доводиться

двоїсто до випадку (а), а для випадку (в) – аналогічно випадку (а).  $\square$

Далі покажемо, що  $\langle \{C, L\}(x') \rangle = \{\Sigma_G, \Lambda_G\}$ . Нехай  $p \in \langle C \rangle$ . З попереднього виконання операції  $ORD(p)$  випливає, що  $p \preceq p^{-1}$  та  $p^{-1} \notin \langle C \rangle$ . З леми 3 випливає, що  $p$  є канонічним словом для  $G$ . Одержали, що  $p$  є канонічним для  $G$  словом або непарної довжини (за лемою 1 у такому випадку  $p^{-1}$  не є канонічним словом), або парної довжини, причому  $p \preceq p^{-1}$ . З урахуванням леми 2,  $p \in \Sigma_G$ , тобто  $\langle C \rangle \subseteq \Sigma_G$ .

Покажемо, що  $\Sigma_G \subseteq \langle C \rangle$ . Нехай  $p \in \Sigma_G$ , вершини  $v', v''$  – твірні для  $p$ ,  $\xi(v') = x'$ ,  $\xi(v'') = x''$ . Нехай для визначеності,  $p = sp_{v'}(sp_{v''})^{-1}$ . За алгоритмом АК  $p \preceq p^{-1}$ , тобто  $sp_{v'}(sp_{v''})^{-1} \preceq sp_{v''}(sp_{v'})^{-1}$ . Оскільки  $\langle \{C, L\}(x') \rangle$  є визначальною парою для  $G$ , то за алгоритмом АП повинно існувати таке слово  $w \in \langle C \rangle \cup \langle L \rangle$ , що ребро  $(v', v'')$  входить до шляху  $v_0w$ . Оскільки за алгоритмом АК  $sp_{v'} \neq sp_{v''}x'$  та  $sp_{v''} \neq sp_{v'}x''$ , то з леми 3 випливає  $w \notin \langle L \rangle$ , та, в залежності від парності числа  $d(p)$ , або  $w = sp_{v'}(sp_{v''})^{-1}$ , або  $w = sp_{v''}(sp_{v'})^{-1}$ . З попереднього виконання операції  $ORD(p)$ , враховуючи  $sp_{v'}(sp_{v''})^{-1} \preceq sp_{v''}(sp_{v'})^{-1}$ , випливає  $w = sp_{v'}(sp_{v''})^{-1}$ , тобто  $w = p$ , тому  $\Sigma_G \subseteq \langle C \rangle$ .

Покажемо, що  $\langle L \rangle = \Lambda_G$ . Оскільки  $\langle \{C, L\}(x') \rangle$  є визначальною парою для  $G$ , то для кожного слова  $q \in \langle L \rangle$  вершина  $v_0q$  є висячою ( $v_0q \neq v_0$ ), та для кожної висячої вершини  $v$  графа  $G$ , відмінної від  $v_0$ , існує хоча б одне слово  $q' \in \langle L \rangle$ , що  $v_0q' = v$ . З урахуванням попереднього виконання операції  $DEL - REP(w)$  та леми 3 одержимо, що  $\langle L \rangle$  складається з усіх таких слів  $w \in \mathcal{V}_G$ , що  $v_0w \neq v_0$  та вершина  $v_0w$  є висячою у графі  $G$ , тому, за алгоритмом АК,  $\langle L \rangle = \Lambda_G$ . Одержали  $\langle \{C, L\}(x') \rangle = \{\Sigma_G, \Lambda_G\}$ .  $\square$

Таким чином, процедура перевірки, чи є задана коректна пара  $\{C, L\}(x')$  визначальною парою для заданого графа  $G$  зводиться до побудови компресії  $\{C, L\}(x')$ , канонічної визначальної пари для  $G$  та порівняння  $\langle \{C, L\}(x') \rangle$  з  $\{\Sigma_G, \Lambda_G\}$ : якщо  $\langle \{C, L\}(x') \rangle = \{\Sigma_G, \Lambda_G\}$ , то  $\{C, L\}(x')$  є визначальною парою для  $G$ ; якщо  $\langle \{C, L\}(x') \rangle \neq \{\Sigma_G, \Lambda_G\}$ , то  $\{C, L\}(x')$  не є визначальною парою для  $G$ . На нашу думку, в обчислювальному сенсі запропонований спосіб є більш простим, ніж побудова графа  $G(\{C, L\}(x'))$  та подальше порівняння  $G(\{C, L\}(x'))$  з  $G$ .

Наведемо приклад визначення, чи є задана коректна пара визначальною для заданого детермінованого графа. Нехай

$C = \{0120, 03530, 03524150, 03114251130, 012035241130\}$ ,

$L = \{035230, 03114230, 0351130123\}$ , та  $\{C, L\}(0)$  – коректна пара. Вихідний граф  $G$  зображено на рис. 3, ініціальна вершина графа  $G$  виділена сірим кольором.

Зробимо компресію вихідної пари. Процес компресії та її результат зображено на рис. 2.

Користуючись алгоритмом АК, знайдемо  $\{\Sigma_G, \Lambda_G\}$ . Одержали  $\Sigma_G = \{0120, 0350, 031150, 0514250\}$ ,  $\Lambda_G = \{023, 05230\}$ . Таким чином,

$\langle \{C, L\}(x') \rangle = \{\Sigma_G, \Lambda_G\}$ , тому  $\{C, L\}(0)$  є визначальною парою для графа  $G$ .

Наведемо важливі властивості компресії, які безпосередньо витікають з теореми 4. Справедливе наступне твердження.

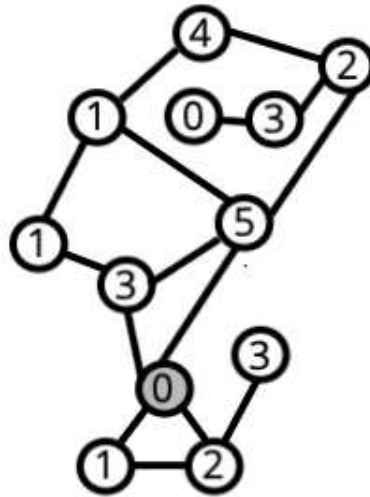


Рис. 3. Вихідний граф  $G$

### Теорема 5.

- 1) Нехай  $\{C, L\}(x')$  – коректна пара. Тоді  $\langle \{C, L\}(x') \rangle$  – коректна пара.
- 2)  $\langle \{\Sigma_G, \Lambda_G\} \rangle = \{\Sigma_G, \Lambda_G\}$ .
- 3) Канонічна визначальна пара  $\{\Sigma_G, \Lambda_G\}$  є найменшою за потужністю та об'ємом кожної компоненти визначальною парою для графа  $G$ .

За наведеними в роботі алгоритмами було розроблено прототип програмної реалізації у вигляді бібліотеки функцій розширюючих можливості пакету NetworkX. Використання цієї бібліотеки дозволяє побудувати граф за заданою парою, канонічну визначальну пару Д-графа, виконати компресію заданої коректної пари та порівняти результат компресії з канонічною визначальною парою заданого графа. Також було досліджено деяку тестову метрику цієї програмної реалізації. При програмній реалізації операції  $PATH - OPT(\{C, L\})$  серед можливих її варіантів, було обрано саме той варіант, тестова метрика якого була найкраща. До складу бібліотеки входить опис нашої програмної реалізації. Ця бібліотека розміщена за адресою [https://github.com/NickIT87/IAMM\\_proceedings](https://github.com/NickIT87/IAMM_proceedings).

### 5. Подальші напрямки дослідження.

Найближчою перспективою досліджень вважається розповсюдження алгоритму компресії на випадок, коли заздалегідь не встановлено, чи є вихідна пара коректною. Також, на нашу думку, наведений алгоритм компресії може бути основою при розв'язанні ряду прикладних задач, пов'язаних з лінгвістичним представленням Д-графів визначальною парою слів, серед яких автори виділяють наступні:

- знаходження необхідних та достатніх умов для множин  $C$  та  $L$ , за яких пара  $\{C, L\}(x')$  є коректною;
- оптимальний вибір ініціальної вершини, за яким метричні властивості компонент канонічної визначальної пари будуть мінімальними;
- ефективна побудова за  $\{\Sigma_G, \Lambda_G\}$  найкоротших за порядком  $\preceq$  шляхів між

двома довільними вершинами графа  $G$ .

## 6. Висновки.

У роботі запропоновано алгоритм компресії, який у випадку коректності вихідної пари, перетворює її в канонічну визначальну пару для Д-графа, для якого вихідна пара є визначальною. Також розглянуто властивості та розроблено прототип програмної реалізації цього алгоритму. Окреслено подальші напрямки дослідження за цією тематикою. Результати дозволяють використовувати нові методи та алгоритми для розв'язання задач аналізу графів з розміченими вершинами.

## Цитована література

1. A.A. Bertossi, M.A. Bonuccelli. Code assignment for hidden terminal interference avoidance in multihop packet radio networks // *IEEE ACM IEEE ACM Transactions on Networking*. – 1995. – Vol. 3, No. 4. – P. 441–449.
2. O.S. Senchenko, M.I. Prytula. The presentation of deterministic and strongly deterministic graphs // *Algebra and Discrete Mathematics*. – 2023. – Vol. 36, Iss. 2. – P. 225–242.
3. S.V. Sapunov Directional Movement of a Collective of Compassless Automata on a Square Lattice of Width 2 // *Cybernetics and Systems Analysis*. – 2024. – Vol. 60, Iss. 6. – P. 899–906.
4. Grunskii I., Mikhaylova I., Sapunov S. Domination on the vertices of labeled graphs // *Algebra and Discrete Mathematics*. – 2012. – Vol. 14, Iss. 2. – P. 174–184.
5. G. Hahn, J. Kratochvíl, J. Širáň, D. Sotteau On the injective chromatic number of graphs // *Discrete Mathematics*. – 2002. – Vol. 256, Iss. 1–2. – P. 179–192.
6. F. Kramer, H. Kramer Ein Färbungsproblem der Knotenpunkte eines Graphen bezüglich der Distanz p // *Revue Roumaine de Mathématiques Pures et Appliquées*. – 1969. – Vol. 14, No. 7. – P. 1031–1038.
7. Grunskii I.S., Senchenko A.S. Properties of systems of defining relations for automata // *Discrete Mathematics and Applications*. – 2004. – Vol. 14, Iss. 6. – P. 593–601.

## References

1. A.A. Bertossi & M.A. Bertossi (1995). Code assignment for hidden terminal interference avoidance in multihop packet radio networks. *IEEE ACM IEEE ACM Transactions on Networking*, 3(4), 441–449. doi: 10.1109/90.413218
2. Senchenko O.S & Prytula M.I. (2023). The presentation of deterministic and strongly deterministic graphs. *Algebra and Discrete Mathematics*, 36(2), 225–242. doi: 10.12958/adm2157
3. S.V. Sapunov (2024). Directional Movement of a Collective of Compassless Automata on a Square Lattice of Width 2. *Cybernetics and Systems Analysis*, 60(6), 899–906. doi: 10.1007/s10559-024-00727-x
4. Grunskii I., Mikhaylova I. & Sapunov S. (2012). Domination on the vertices of labeled graphs. *Algebra and Discrete Mathematics*, 14(2), 174–184.
5. G. Hahn, J. Kratochvíl, J. Širáň & D. Sotteau (2002). On the injective chromatic number of graphs. *Discrete Mathematics*, 256(1–2), 179–192. doi: 10.1016/S0012-365X(01)00466-6
6. F. Kramer, H. Kramer (1969). Ein Färbungsproblem der Knotenpunkte eines Graphen bezüglich der Distanz p. *Revue Roumaine de Mathématiques Pures et Appliquées*, 14(7), 1031–1038. doi: 10.13140/RG.2.2.24729.83047
7. Grunskii I.S. & Senchenko A.S. (2004). Properties of systems of defining relations for automata. *Discrete Mathematics and Applications*, 14(6), 593–601. doi: 10.1515/1569392043272458

O.S. Senchenko, M.I. Prytula

The properties of compression of defining pair for deterministic graphs.

In this article, we consider an algorithm for compressing a pair that is defining for a deterministic graph and the properties of this algorithm. Deterministic graphs are actively used as a model of many software objects, information processes, digital network topology, etc. The proposed algorithm is associated with the linguistic presentation of deterministic graphs by two sets of words of a special kind (the «pair»): the first set describes the graph cycles, and the second set describes its leaf vertices. The basis of the linguistic representation is the algorithm, which, given a pair, either constructs a deterministic graph for which this pair is the presentation or informs that it is impossible to do so. We have also developed an algorithm for constructing a special pair for any deterministic graph, called a canonical pair, and proved that the canonical pair is a presentation of this graph. In this article, we propose elementary transformations of a pair that is a presentation of a certain deterministic graph (the so-called «correct pair»). As a result of these transformations, the graph, which is a presentation by the original pair, is isomorphic to the graph, which is a presentation by the result of elementary transformations with the original pair. On the basis of these elementary transformations, we develop an algorithm for compressing the correct pair. It is proved that the result of compression is equal to the canonical pair of the graph, which is the presentation by the original pair; this is the main result of this article. The properties of this algorithm are also presented, in particular, it is shown that the smallest pair in terms of power and volume of each component, which is a presentation of an arbitrary deterministic graph, is its canonical pair. Based on the compression algorithm, a solution to the problem of pair characterization for a graph is proposed: the task of determining whether a given pair is a presentation of a given deterministic graph, in the case of correctness of the pair, is reduced to constructing a compression of the given pair, the canonical pair for a given graph, and comparing these pairs. If the above pairs are equal, then the given pair is a presentation of the given deterministic graph, otherwise the given pair is not a presentation of the given deterministic graph. In our opinion, in a computational sense, the proposed method is simpler than constructing a graph based on a given pair and then comparing it with a given graph. Further directions of research on this topic are also outlined. The results will allow the use of new methods and algorithms to solve problems of analyzing graphs with labeled vertices.

**Keywords:** *compression, deterministic graphs, presentation, defining pair.*

Інститут прикладної математики і механіки НАН України,  
Слов'янськ  
*Senchenko.a76@gmail.com, elanir358@gmail.com*

*Отримано 28.03.25*