Міністерство освіти і науки України

НАЦІОНАЛЬНИЙ УНІВЕРСИТЕТ
“КИЄВО-МОГИЛЯНСЬКА АКАДЕМІЯ”

Кафедра інформатики
Факультет інформатики

Курсова робота на тему:
Групоїди на графах і орграфах

Керівник курсової роботи:
к. ф.-м. н. *Козеренко С.О.*
(прізвище та ініціали)

_____
(підпис)
“____”_____ 2024 р.

Виконав студент
3-го року навчання спеціальності
121 “Інженерія програмного забезпечення”
Первушин Кирило Ігорович
(ПІБ)

Київ – 2024

Анотація

У роботі розглянуті властивості однієї бінарної операції на множині вершин дерева. Доведено, що групоїд із такими властивостями однозначно відповідає деякому дереву. Надана якісна та кількісна характеризація комутативних пар та асоціативних трійок бінарної операції. Запрограмовано перетворення між комп'ютерним представленням графа й отриманої алгебраїчної структури.

Ключові слова: T-групоїд, дерево, групоїд, твірна множина, гомоморфізм групоїдів, код Прюфера.

Зміст

1 Графік роботи

Номер	Назва етапу курсової	Термін виконання етапу	Примітка
1.	Отримання теми курсової роботи.	вересень	
2.	Ознайомлення з темою курсової.	вересень	
3.	Розробка плану та структури роботи.	жовтень	
4.	Робота з науковою літературою, опис основних означень теорії графів, ознайомлення з метричною теорією графів.	листопад	
5.	Дослідження основних властивостей алгебраїчних структур та можливості їх визначення на графах.	грудень-січень	
7.	Робота над текстовим оформленням результатів. Вивчення роботи в текстовому процесорі LaTeX.	лютий-квітень	
8.	Попередній аналіз курсової. Виправлення помилок.	квітень	
9.	Захист курсової. роботи.	24.05.2023	

Вступ

Графи виникають у різноманітних задачах комп'ютерних наук, планування та логістики як гнучкі математичні структури, за допомогою яких можна точно описати зв'язки між об'єктами, абстрагуючись від маловажливих умов (таких, як відстані та кути в метричному просторі). Поряд із високим інтересом до вивченого питання, математична складова теорії графів спонукає науковців звернутись до інших розділів математики за аналогіями та відповідями на поставлені питання. Хоч за природою графи є топологічними структурами, для формулювання та вирішення задач стають у нагоді теорія множин, аналіз, комбінаторика, теорія алгоритмів і обчислювальності тощо.

Ідеєю цієї роботи є спроба вивчення графів засобами абстрактної алгебри, тобто проведення паралелей між властивостями бінарних операцій на множині та операціями над графами. Ідеєю для роботи послугувала стаття **[1]**, де введено поняття геодетичного графа та розглянуто властивість однієї бінарної операції на множині вершин такого графа. Зокрема, доводиться, що клас скінченних дерев із теорії графів еквівалентний певній алгебраїчній структурі із бінарною операцією, що задовольняє деяким умовам.

У даній роботі переглянуто зв'язок між графами та групоїдами та зроблено подальші дослідження та висновки з еквівалентності цих класів. Для графів досліджуються питання, що класично постають при вивченні алгебраїчної структури: дослідження її підструктур, твірних підмножин, гомоморфізмів. Підняте питання комп'ютерного зображення розглянутих структур, як графів, так і групоїдів, та наведено приклад переходу між цими репрезентаціями засобами мови системного програмування Rust.

2 Означення

Означення 2.1. *Неорієнтований граф – це пара* $G = (V, E)$*, де* $V = V(G)$ *– множина вершин,* $E = E(G) \subset V \times V = \{\{a, b\} |\ a, b \in V\}$ *– множина ребер.*

Означення 2.2. *Орієнтований граф – це пара* $G = (V, E)$*, де* $V = V(G)$ *– множина вершин,* $E = E(G) \subset V \times V = \{(a, b) |\ a, b \in V\}$ *– множина дуг.*

Означення 2.3. *Степенем вершини* u *неорієнтованого графа* G *називають кількість вершин* $v \in G$ *таких, що* $u \neq v$ *та* $uv \in E(G)$*. У роботі використовуватиметься позначення* $d_G(u)$*.*

Означення 2.4. *Шлях між парою вершин* $x, y \in V(G)$ *– це послідовність вершин* $v_0, v_1, \ldots, v_n$*, де* $x = v_0$ *– початок шляху,* $y = v_n$ *– кінець шляху,* $\forall i = 0 \ldots n-1 :\ v_i v_{i+1} \in E(G)$*. Шлях є циклом, якщо його початок та кінець співпадають.*

Означення 2.5. *Граф називається зв'язним, якщо між будь-якою парою вершин існує шлях.*

Означення 2.6. Дерево *– це зв'язний неорієнтований граф, що не містить цикла.*

В даній роботі розглядатимемо тільки скінченні дерева.

Нехай G - зв'язний граф. Позначимо $d_G(u, v)$ як відстань між вершинами $u, v \in G$. Для кожної впорядкованої пари $u,\ v \in G,\ u \neq v$ позначимо

$$A_G(u, v) = \{w \in V(G);\ d_G(u, w) = 1\ and\ d_G(w, v) = d_G(u, v) - 1\}.$$

Означення 2.7. *Граф* G *називається* геодетичним*, якщо він зв'язний та між кожними двома різними вершинами існує єдиний найкоротший шлях, тобто*

$$|A_G(u, v)| = 1\ \forall u, v \in V(G).$$

Означення 2.8. *Метричним відрізком* $[u, v]_G$ *між вершинами* $u,\ v \in V(G)$ *зв'язного графа називатимемо підмножину вершин* w*, таких що*

$$d_G(w, u) + d_G(w, v) = d_G(u, v).$$

Означення 2.9. *Якщо* $uv \in E(G)$, *то півпростором* $W(u,v) \subset V(G)$ *називається множина*

$$\{w \in V(G) \mid d_G(w,u) < d_G(w,v)\}.$$

Означення 2.10. *Якщо* $G = (V(G), E(G))$ *граф, то графом, породженим множиною* $A \in V(G)$ *називають граф* $G[A]$ *із* $V(G[A]) = A$, $E(G[A]) = \{\{u,v\} \in E(G) \mid u,v \in A\}\}$.

Означення 2.11. *У графі* G *підмножина* $A \in V(G)$ *називається зв'язною, якщо породжений нею підграф* $G[A]$ *зв'язний.*

3 Індукована операція геодетичного графа

Означення 3.1. *Індукованою операцією геодетичного графа G називатимемо таку бінарну операцію $+ : V(G) \times V(G) \to V(G)$, що $u + u = u$ та $u + v$ $(u \neq v)$ - єдиний елемент $A(G)$.*

Геодетичність графа гарантує, що індукована операція однозначна, тож є коректною бінарною операцією на множині вершин.

Лема 3.2. *Нехай T - дерево, $+$ є індукованою операцією на T. Нехай $V = V(T)$. Тоді $+$ задовольняє таким аксіомам:*
(A) $(u + v) + u = u$ (для всіх u, $v \in V$);
(B) якщо $(u + v) + v = u$, то $u = v$ (для всіх u, $v \in V$);
(C) якщо $u \neq u + v = v \neq u + w$, то $v + w = u$ (для всіх u, v, $w \in V$).

Доведення. □ ***(A)**: Якщо $u = v$, то з ідемпотентності операції $+$ слідує правильність твердження. Інакше, $u + v$ та u з'єднані ребром, тоді $A_T(u + v, u) = \{u\}$.**(B)**: Нехай $u \neq v$. Тоді $d_T((u + v) + v, u) \geq 1$, що протирічить $(u + v) + v = u$. Тож $u = v$.**(C)**: Із $u \neq u + v = v$ слідує $d_T(u, v) = 1$ та $(u, v) \in E(T)$. Доведемо, що w лежить у півпросторі $W(u, v)$. Справді, так як T – дерево, то найкоротший шлях від u до вершини x підпростору $W(v, u)$ лежить через v, тож $u+x = v$ $\forall x \in W(v, u)$. Значить $w \notin W(v, u)$ та $w \in W(u, v)$. Але тоді найкорошший шлях від v до w лежить через u, тобто $v + w = u$.* ■

Нехай $+$ є бінарною операцією на скінченній непорожній множині V та задовольняє аксіоми (A), (B), (C). Тоді пару $(V, +)$ називатимемо **Т-групоїдом**. Для Т-групоїда $\Gamma = (V, +)$ позначатимемо $V(\Gamma) = V$. У цьому розділі буде доведено, що кожному Т-групоїду відповідає дерево, тобто між деревами та Т-групоїдами існує взаємно-однозначна відповідність.

Лема 3.3. *Нехай $\Gamma = (V, +)$ – Т-групоїд. Тоді:*

1. $u + v = v$ тоді й тільки тоді, коли $v + u = u$ $\quad \forall u, v \in V$;

2. $u + v = u$ тоді й тільки тоді, коли $u = v$ $\quad \forall u, v \in V$.

Доведення. □ *1. У частині (A) **Леми 2.3** покладемо $u + v = v$, тоді $v+u = u$. Замінюючи ролі u та v, доведемо, що з $v+u = u$ слідує $u+v = v$.*

*2. У частині (B) **Леми 2.3** покладемо $u + v = u$, тоді імплікація приймає вигляд $u+v = u \Rightarrow u = v$, тож $u = v$. Навпаки, якщо $u = v$, то з доведеного пункту 1 цієї леми слідує ідемпотентність $+$; $(\forall u,\ u+u = u)$, тож $u + v = u + u = u$, що і треба було довести.* ■

Означення 3.4. *Нехай* $\Gamma = (V_\Gamma, +)$ *– T-групоїд. Граф* G *називається асоційованим із* Γ*, якщо* $V(G) = V_\Gamma$ *та*

$$E(G) = \{\{u, v\} \mid u, v \in V(G),\ u + v = v \neq u\}.$$

За пунктом 1 ***Леми 2.3***, існує тільки один граф, асоційований із заданим T-групоїдом. Взаємна однозначність між T-групоїдами та деревами задається наступною теоремою.

Теорема 3.5. *Нехай* $\Gamma = (V_\Gamma, +)$ *– T-групоїд,* $G = (V, E)$ *– граф, асоційований до* Γ*. Тоді* G *– дерево і* $+$ *– індукована операція на* G*.*

Доведення. □ *Доведемо спочатку, що якщо* H *– компонента* G*, то*

$$A_H(x, y) = \{x + y\}\ \forall x, y \in H.$$

Якщо H *тривіальна, тобто містить лиш одну вершину, тоді твердження очевидне. Нехай* $|V(G)| \geq 2$*. Виберемо будь-які дві різні вершини* $x, y \in V(G)$ *та позначимо* $k = d_H(x, y)$*,* $k \geq 1$*. Скористаємось індукцією за* k*. Для* $k = 1$ *маємо* $xy \in E(G)$*, тоді* $x + y = y$*,* $y + x = x$*, значить* $A_H(x, y) = \{x + y\}$ *і* $A_H(y, x) = \{y + x\}$*. Припустимо тепер, що* $k \geq 2$ *і* $\forall u, v \in V(H)$ *таких, що* $d_H(u, v) = k - 1$ *маємо* $A_H(u, v) = \{u + v\}$*. Тоді виберемо* $z \in A_H(x, y)$*.* $xz \in E(G)$ *та* $d_G(z, y) = k - 1$*, значить* $z + x = x \neq z + y$*, тож, за частиною (C)* ***Леми 2.3****, маємо* $x + y = z$ *та* $A_H(x, y) = \{x + y\}$*.*

Маємо, що кожна компонента породженого графа є геодетичним підграфом. Припустимо, що в породженому графі існує цикл непарної довжини. Тоді $\exists x, y, z \in V(G)$ *такі, що* $d_G(x, y) = 1$ *та* $d_G(x, z) = d_G(y, z)$*. Значить* $x + y = y \neq x + z$*, тож, за частиною (C)* ***Леми 2.3****, маємо* $y + z = x$*, з чого слідує* $d_G(y, z) = d_G(x, z) + 1$*, суперечність. Тож припущення про існування цикла непарної довжини неправильне і асоційований граф є лісом.*

Доведемо, що цей ліс зв'язний. Для довільної пари різних елементів $x, y \in \Gamma$ *побудуємо послідовність* $x_0 = x$*,* $x_n = x_{n-1} + y$*. Асоційований граф не містить циклів, тож всі* x_n*,* $n \geq 0$ *різні. Тоді послідовність скінченна та* $x_m = y$ *для декотрого* $m \geq 1$*. Значить* x *та* y *лежать в одній компоненті зв'язності. Так як вибір вершин був довільним, асоційований граф зв'язний, тобто є деревом, що і треба було довести.* ■

4 Алгебраїчні властивості індукованої операції

Розглянемо властивості індукованої операції T-групоїда та дослідимо, яким умовам має задовольняти асоційований граф, щоб ці властивості виконувались.

Означення 4.1. *Бінарна операція* $+ : X \times X \to X$, *задана на деякій множині* X, *називається комутативною, якщо* $\forall a, b \in X,\ a + b = b + a$.

Очевидно, що в T-групоїді $+$ в загальному випадку не є комутативною.

Твердження 4.2. *На T -групоїді для вершин* u, v *справедливо, що* $u + v = v + u$ *тоді і тільки тоді, коли* $u = v$ *або* $d_T(u,\ v) = 2$.

Доведення. □ *Якщо* $u = v$, *то очевидно* $u+v = v+u$. *Якщо ж* $d_T(u,\ v) = 2$, *то помітимо, що* $A_G(u, v) = \{w\}A_G(v, u)$, *де* w - *єдина вершина така, що* $d_G(u, w) = d_G(w, v)$, *тоді* $u + v = v + u = w$. *Навпаки, нехай* $u \neq v$ *та* $u + v = v + u = w$, *тоді* $d_G(u, w) = d_G(w, v)$, *звідки* $d_G(u, v) \leq 2$. *Але при* $d_G(u, v) = 1$ *існує ребро між* u *та* v, *тоді* $u+v = v \neq v+u = u$. *Значить,* $d_G(u, v) \leq 2$. ■

Підрахунок кількості комутатуючих пар T-групоїда привів до наступного результату.

Означення 4.3. *Першим загребським індексом* $M_!(G)$ *неорієнтованого графа* G *називають значення* $M_1(G) = \sum_{u \in V(G)} d_G^2(u)$.

Твердження 4.4. *Кількість (невпорядкованих) комутуючих пар відносно* $+$ *на дереві* T *з* $|V(T) > 1|$ *дорівнює* $1 + \frac{1}{2}M_1(T)$.

Доведення. □ *Доводитимо за допомогою індукції за кількістю вершин у дереві, починаючи з двох вершин. Для* $|V(T) = \{u, v\}|$ *маємо дві комутативні пари* $u+u = u$ *та* $v+v = v$, *і* $1+\frac{1}{2}M_1(T) = 1+\frac{1^2+1^2}{2} = 2$. *Нехай для деякого дерева з* k *вершинами кількість комутативних пар дорівнює, згідно з припущенням,* $K(T) = 1 + \frac{1}{2}M_1(T)$. *Тоді при додаванні до дерева* T *однієї вершини (аби граф лишився деревом, нова вершина має бути з'єднана ребром із рівно однією існуючою вершиною - назвемо цю вершину* x, *а нову –* y*) Загребський індекс стає* $M_1(T+\{y\}) = M_1(T)+1+(d_T(x)+1)^2-d_T^2(x) = M_1(T) + 2(d_T(x) + 1)$. *Порахуємо нові комутативні пари: одна пара за рахунок ідемпотентності* $+ : y + y = y$ *та* $d_T(x)$ *пар* $\{y, w\}$, *де* w *така що*

$\{x, w\} \in E(T)$, *бо в новому дереві* $T + \{y\}$ *маємо* $y + w = w + y = x$, *тож загалом:*

$$K(T + \{y\}) = K(T) + 1 + d_T(x) = 1 + \frac{M_1(T) + 2(1 + d_T(x))}{2} = \\ = 1 + \frac{M_1(T + \{y\})}{2}.$$

■

Тепер розглянемо, коли операція + T-групоїда є асоціативною.

Означення 4.5. *Асоціативною трійкою для бінарної операції* $+ : X \times X \to X$, *заданої на деякій множині* X, *називається впорядкована трійка елементів* $(a, b, c) \in X$ *така, що* $a + (b + c) = (a + b) + c$.

Дослідження асоціативних трійок цікавить нас, як основа для побудови складніших алгебраїчних конструкцій.

Теорема 4.6. *Нехай* $T = (V, E)$ *– дерево,* $+$ *– індукована операція на* T. *Впорядкована трійка* $(a, b, c) \in V^3$ *є асоціативною для* $+$ *тоді й тільки тоді, коли* $a = b$, *або* $a + b = c$, *або* $b + c = a$.

Доведення. □ *Розглядатимемо кілька варіантів взаємного розташування вершин* $a, b, c \in V(T)$ *та визначатимемо, за яких умов така конфігурація є асоціативною. Таким чином, розгляд всіх можливих конфігурацій дасть необхідну характеризацію.*

1. *Якщо* $a = b = c$, *то асоціативність очевидна.*

2. *Якщо* $a = b \neq c$, *то* $a + (b + c) = a + (a + c) = a + c = (a + a) + c = (a + b) + c$, *тобто асоціативність виконується завжди.*

3. *Якщо* $a = c \neq b$, *то* $(a + b) + c = (a + b) + a = a$ *і, за частиною (B)* ***Леми 2.2***, $a+(b+a) = a$ *еквівалентно* $b+a = a$, *тобто* $\{a, b\} \in E(T)$.

4. *Якщо* $a \neq b = c$, *то умова асоціативності перетворюється на* $a + (b + b) = (a + b) + b$, *що, за частиною (B)* ***Леми 2.2***, *знову еквівалентно* $\{a, b\} \in E(T)$.

5. *Якщо* $a \neq b \neq c$, *то позначимо* $u = a + c$, $v = a + b$ *і помітимо, що* $w = a + (b + c)$ *суміжна з* a *або рівна їй. Так як* u, v *суміжні з* a, *то або* $u = v$, *або, за частиною (C)* ***Леми 2.2***, *з* $u \neq v$ *отримаємо* $u + v = v + u = a$ *і* $v + c = (a + b) + c = a$.

(а) Якщо $u = v$, *то* $(a + b) + c = (a + c) + c$ *рівна* $a + (b + c)$ *тоді й тільки тоді, коли* $a + c = c$ (*тобто* $\{a, c\} \in E(T)$). *Але* $a + c = a + b$, *тому асоціативність еквівалентна* $a + b = c$.

(б) Якщо ж $u \neq v, (a + b) + c = a$, *то, за частиною (2)* ***Леми 2.3***, $a + (b + c) = (a + b) + c$ *тоді й тільки тоді, коли* $b + c = a$.

Об'єднуючи випадки 1 та 2, маємо асоціативність при $a = b$, *випадок 3 разом із 5(б) дає асоціативність при* $b + c = a$, *випадок 4 разом із 5(а) дає асоціативність при* $a + b = c$. ■

Зауваження. *З доведення* ***Теореми 3.6*** *видно, що впорядкована трійка* $(a, b, c) \in V^3(T)$ *є асоціативною відносно* $+$ *тоді й тільки тоді, коли* $a = b$, *або* $c \in W_T(a, b)$, *або* $b \in W_T(c, a)$.

Із теореми 1 одразу можна обчислити кількість таких трійок, яка як виявляється, не залежить від структури самого дерева.

Наслідок 4.6.1. *Кількість асоціативних трійок відносно* $+$ *на* n *– вершинному дереві дорівнює* $3n^2 - 2n$.

Доведення. □ *Позначимо* $|n = V(T)|$ *попереднього* ***Зауваження*** *порахуємо: кількість трійок* $(a, b, c) \in V^3(G)$ *таких, що* $a = b$ – n^2, *Для кожного ребра* $\{u, v\} \in E(T)$, *яких у дереві* $n - 1$*:*

- *якщо* $a = u, b = v$ – $|W_T(u, v)|$ *варіантів для* c*;*
- *якщо* $a = v, b = u$ – $|W_T(v, u)|$ *варіантів для* c*;*

Так як у дереві $|W_T(u, v)| + |W_T(v, u)| = n$, *то варіант* $b + c = a$ *дає загалом* $n^2 - n$ *асоціативних трійок. Аналогічно* $a + b = c$ *дає ще* $n^2 - n$ *трійок. Об'єднуючи ці три взаємовиключні випадки, маємо* $3n^2 - 2n$ *асоціативних трійок.* ■

5 Підгрупоїди та твірна множина

При вивченні алгебраїчних структур природньо постають питання їхньої складу в термінах підмножин елементів. У часності, певна підмножина елементів може сама бути коректною алгебраїчною структурою відносно тої ж бінарної операції тоді, коли ця підмножина замкнена відносно цієї операції. У теорії груп такі підмножини називаються **підгрупами**, ми ж у цьому розділі розглянемо підгрупоїди, які визначаються абсолютно аналогічно. Також для алгебри цікаво в множині знайти підмножину таку, що, ітеруючи бінарну операцію між елементами знайденої підмножини, можна отримати будь-який елемент початкової множини – такі підмножини називаються **твірними**. Вивчення властивостей алгебраїчної структури можна звести до вивчення елементів твірних підмножин – тут можна провести аналогію із базисом векторного простору в лінійній алгебрі. Нижче ми розглянемо підгрупоїди T-групоїдів та їхні твірні множини.

Означення 5.1. *Множина $H \subset G$ називається підгрупоїдом групоїда $(G, \oplus)$, якщо $\forall a, b \in H,\ a \oplus b \in H$.*

Інакше кажучи, H замкнена відносно $\oplus$. Підгрупоїд сам по собі також є групоїдом. Весь групоїд завжди є своїм підгрупоїдом; якщо $\oplus$ ідемпотентна, як у випадку T-групоїдів, то одноелементні підмножини також є підгрупоїдами. Для доведення характеризації підгрупоїда T-групоїда нам знадобиться наступне твердження:

Твердження 5.2. *У графі G підмножина $A \subset V(G)$ зв'язна тоді й тільки тоді, коли $\forall u, v \in A$ існує шлях між u та v такий, що кожна вершина на цьому шляху також лежить в A.*

Доведення. □ *Нехай $A \subset V(G)$ зв'язна в G. Тоді породжений підграф $G[A]$ зв'язний. Значить, $\forall u, v \in G[A]$ існує шлях у $G[A]$ між u та v. За означенням породженого підграфа, кожна вершина цього шляху також лежить в A. Інакше, нехай $\forall u, v \in A$ існує шлях в $G[A]$ між u та v. Тоді $G[A]$ зв'язний і A зв'язна.* ■

Твердження 5.3. *Нехай T – дерево. Тоді $A \subset V(T)$ є підгрупоїдом відносно $+$ тоді й тільки тоді, коли A – зв'язна множина в T.*

Доведення. □ *Нехай A – зв'язна підмножна T. Тоді $\forall u, v \in A$, $u + v$ лежить на шляху між u та v, який єдиний, і, значить, $u + v \in A$, так як A – зв'язна множина. Тож A замкнена відносно $+$ і $(A, +)$ – підгрупоїд.*

Навпаки, нехай $(A, +)$ – підгрупоїд в T. Тоді $\forall u, v \in A$ можна побудувати скінченну послідовність $u_0 = u$, $u_n = u_{n-1} + v$ таку, що $u_N = v$. Ця послідовність визначає шлях в T між u та v. Так як A – підгрупоїд, всі елементи цієї послідовності лежать в A, тож в графі T вона зв'язна за доведеним вище ***Твердженням 4.2***. ■

Так як зв'язні множини в дереві є його піддеревами, і, навпаки, кожне піддерево зв'язне, то підгрупоїди T-групоїда це в точності піддерева асоційованого йому дерева.

Далі, нас цікавлять твірні підмножини T-групоїда.

Означення 5.4. *Підмножина A групоїда $(G, \oplus)$ називається твірною, якщо найменшим підгрупоїдом у G, що містить A, є весь G.*

Наступне твердження дає необхідну та достатню умову для підмножини $A \subset V(T)$ бути твірною:

Твердження 5.5. *Нехай T – дерево. Тоді $A \subset V(T)$ є твірною відносно $+$ тоді й тільки тоді, коли A містить всі висячі вершини T.*

Доведення. □ *Нехай $A \subset V(T)$ є твірною. Зауважимо, що якщо $u \in V(T)$ висяча, то єдиними способами отримати u за допомогою $+$ є $u = u + u$ та $u = x + u$ де x суміжна з u в T, бо перший операнд $+$ завжди суміжний з результатом. Тож якщо множина не містить висячих, то вона не є твірною – значить, A містить всі висячі вершини T. Навпаки, якщо A містить всі висячі вершини T, то $\forall x \in V(T)$ існують такі $u, v \in A$, що x лежить на найкоротшому шляху між u та v. Значить, x можна отримати, ітеруючи $+$ на $\{u, v\} \subset V(T)$. Тож A є твірною множиною для графа T.* ■

6 Гомоморфізми

Вивчення гомоморфізмів – відображень, що зберігають структуру – займає ключову роль в абстрактній алгебрі, лінійній алгебрі та їхніх підрозділах. Побудова гомоморфізма може вказати на важливі зв'язки між складними та простішими алгебраїчними структурами, дозволяє дослідити поведінку їх елементів та підструктур, проводити кількісний аналіз. Гомоморфізми також визначені на графах, і дозволяють змоделювати багато задач цієї різноманітної теорії (наприклад, задача про розфарбування еквівалентна знаходженню хоча б одного гомоморфізма з заданого графа на повний неорієнтований граф). У даному розділі визначаються гомоморфізми між T-групоїдами та досліджується, коли гомоморфізм дерев є гомоморфізмом їхніх T-групоїдів і навпаки.

Означення 6.1. *Гомоморфізмом між двома групоїдами* $(X, \oplus)$ *та* $(Y, \otimes)$ *називається відображення* $f : X \to Y$ *таке, що* $f(a \oplus b) = f(a) \otimes f(b)$ *для всіх* $a, b \in X$.

Означення 6.2. *Відображення* $f : V(G) \to V(H)$ *між графами* G, H *називається гомоморфізмом, якщо для всіх ребер* $uv \in E(G)$ *виконується* $f(u)f(v) \in E(H)$.

Наступний результат пов'язує гомоморфізми T-групоїдів із гомоморфізмами дерев:

Теорема 6.3. *Відображення* $f : V(T_1) \to V(T_2)$ *між двома деревами* T_1, T_2 *є гомоморфізмом між їхніми T-групоїдами тоді й тільки тоді, коли* f *– постійне або є ін'єктивним гомоморфізмом цих дерев.*

Доведення. *Якщо відображення* $f : V(T_1) \to V(T_2)$ *таке, що* $\forall x, y \in V(T)$, $f(x) = f(y) = u \in V(T_2)$, *то* $f(x + y) = f(x) + f(y) = u + u = u$, *тож* f*-гомоморфізм. Нехай тепер* f *є ін'єктивним гомоморфізмом. Тоді зокрема* f*-гомоморфізм. Нехай* $f : V(T_1) \to V(T_2)$ *гомоморфізм. Розглянемо ребро* $\{a, b\} \in E(G)$. *Його образом під* f *буде або одна вершина, або ребро в* T_2. *Значить, весь* $f[T_1]$ *- або одна вершина в* T_2, *або підграф, ізоморфний* T_1.

7 Код Прюфера та матриця Келі T-групоїда

У цьому розділі представлено огляд цікавого способу комп'ютерного зображення дерев - кода Прюфера, і перетворення між кодом Прюфера дерева та зображенням алгебраїчних структур - матрицією Келі. Як мову програмування я обрав Rust - відносно нову мову системного програмування з безпечним керувнням пам'яттю та конкурентністю. Такий вибір обумовлений тим, що ця мова є компільованою, за швидкісними параметрами не поступається C, має повний функціонал доступу до ресурсів ядра та низькорівневим регулюванням пам'яті, а також містить багато семантичних конструкцій із функціонального програмування (зокрема розвинену типізацію), навіяних чисто функціональною мовою Haskell. Отже, Rust є ідеальним кандидатом для побудови складних обчислювальних бібліотек, математичних фреймворків, засобів обробки відео, аудіо та зображень.

Кодом Прюфера називається компактний спосіб комп'ютерного представлення дерев, що базується на визначальних властивостях саме цього класу неорієнтованих графів, а саме зв'язності, відсутності циклів, а також відомому факті, що кожне дерево із n вершинами містить в точності $n+1$ ребер. Код Прюфера кодує дерево з n вершинами в масив цілих чисел довжини $n-2$ так, що за цим масивом можна однозначно відновити структуру дерева.

Алгоритм побудови кода Прюфера наступний:

1. Кожній вершині заданого графа співставимо унікальне число від 1 до $|V(G)|$.

2. На кроці i, виберемо вершину з найменшим індексом a. Видалимо її з дерева та в наступним (i – тим)значенням коду запишемо номер єдиної (бо обрана вершина висяча) вершини залишкового дерева, що суміжна з обраною.

3. Повторюєил, доки не лишиться тільки дві вершини (їхніми номерами будуть n та $n-1$).

Наведемо код перетворення кода Прюфера дерева на таблицю суміжності вершин.

```
fn convert_prufer_to_lists(prufer: Vec<usize>) -> Vec<Vec<usize>> {
    let n = prufer.len() + 2; // number of vertices in a tree
    // Array representing degree of a vertex minus its degree in a
    // part of graph already constructed. (i.e. number of edges
    // still to be connected here). On each step, this number will
```

```
    // decrease. Once initialised, the array shows valid degree of
    // each vertex, computed using information from Prufer's code.
    let mut degree = Vec::<usize>::with_capacity(n);
    for _i in 0..n {
        degree.push(1);
    }
    for c in &prufer {
        degree[*c] = degree[*c] + 1;
    }

    // Now lets build data structures for graph representation.
    // Here these are adjacency lists.
    let mut graph = Vec::with_capacity(n);
    for _i in 0..n {
        graph.push(Vec::<usize>::new());
    }

    for c in &prufer {
        for i in 0..n {
            let j = *c;
            if degree[i] == 1 {
                graph[j].push(i);
                graph[i].push(j);
                degree[j] = degree[j] - 1;
                degree[i] = degree[i] - 1;
                break;
            }
        }
    }

    let mut u : usize = 0;
    let mut v : usize = 0;
    for i in 0..n {
        if degree[i] == 1 {
            if u == 0 {
                u = i;
            } else {
                v = i;
            }
        }
    }
    graph[u].push(v);
    graph[v].push(u);
    degree[u] = degree[u] - 1;
    degree[v] = degree[v] - 1;
    graph
}

```

Природнім представленням алгебраїчних структур є **таблиця Келі**. Для алгебраїчної структури, заданої на множині скінченної потужності n, таблицею Келі називають квадратну матрицю розмірності $n*n$, де кожному рядку та кожному стовпцю відповідає елемент цієї множини, а елемента-

ми матриці є результат виконання бінарної операції на елементі, який відповідає стовпцю, та елементі, який відповідає рядку. Тобто, для операції, заданої в мультиплікативному стилі, таблиця відображає всі можливі добутки. Таблиці Келі слугують гарним описом для груп і догрупових структур (напівгруп, моноїдів, групоїдів і т. д.). За такою матрицею можна швидко визначити багато властивостей операції: її комутативність, ідемпотентність, наявність одиниць та нейтральних елементів для конкретних елементів чи глобально в множині; перевірити, чи є дана структура квазігрупою (тобто чи розв'язні рівняння $ax = b$ та $ya = b$ відносно x, y для довільних елементів a, b – достатньо перевірити, чи є таблиця латинським квадратом). Асоціативність за таблицею Келі перевірити в загальному випадку складно: прямий алгоритм із перебором усіх трійок потребує $\mathcal{O}(n^3)$ операцій, проте існують складніші алгоритми для цієї задачі. Критерій Лайта дозволяє визначити асоціативність за час $\mathcal{O}(m * n^2)$, де m – кількість елементів деякої власної твірної множини цієї операції.

Перетворення між отриманим із кода Прюфера представленням дерева у вигляді списків суміжності та матрицею Келі, що зображає групоїд, робитимемо за допомогою пошука в ширину (**breadth-first search**), що його проводитимемо по черзі з кожної вершини (у результаті складність алгоритма буде $\mathcal{O}(n^2)$). Відвідуючи кожну вершину v, відмінну від кореня пошуку u, просто позначатимемо в таблиці Келі результат виконання $u + v$ як першу вершину на шляху від кореня до v. Таким чином, кожна ітерація пошуку в ширину заповнюватиме рядок та стовпчик таблиці Келі, що відповідають u. Хоч автор допускає можливі покращення цього наївного алгоритму, кращої асимптотики досягти не вийде, адже заповнення квадратної таблиці завжди займе як мінімум $\mathcal{O}(n^2)$.

```rust
use std::collections::VecDeque;

fn bfs(adj_list: &Vec<Vec<usize>>, root: usize) -> Vec<usize> {
    let mut queue = VecDeque::new();
    let mut visited = vec![false; adj_list.len()];
    let mut results : Vec<usize> = vec![usize::MAX; adj_list.len()];

    queue.push_back(root);
    visited[root] = true;
    results[root] = root; // u + u = u

    while let Some(v) = queue.pop_front() {
        for &w in &adj_list[v] {
            if !visited[w] {
                queue.push_back(w);
                visited[w] = true;
                if v == root {
```

```
                    results[w] = w;
                } else {
                    results[w] = results[v]; // u + w = u + v
                }
            }
        }
    }
    results
}

fn convert_lists_to_cayley(graph : Vec<Vec<usize>>) -> Vec<Vec<usize>> {
    let mut cayley = vec![Vec::new(); graph.len()];
    for root in 0..graph.len() {
        cayley[root] = bfs(&graph, root);
    }
    cayley
}

```

Таким чином, із кода Прюфера отримаємо таблицю $n*n$ - таблицю Келі відповідного T-групоїда. Дла повноти додаю тестувальну функцію:

```
fn main() {
    let prufer = vec![3, 3, 3, 4];
    let graph = convert_prufer_to_lists(prufer);
    let table = convert_lists_to_cayley(graph);
    for i in 0..table.len() {
        for j in 0..table[i].len() {
            print!("{} ", table[i][j]);
        }
        print!("\n");
    }
}

```

8 Висновки

Ця робота є спробою пов'язання двох розділів чистої математики, що знайшли широкі застосування в усіх областях комп'ютерних наук: теорії графів та абстрактної алгебри. Детально досліджено алгебраїчні властивості однієї бінарної операції на множині вершин графа та групоїд, який вона визначає - T-групоїд. Показано, що такі операції можуть мати рівносильні визначення в термінах графів та у вигляді алгебраїчних співвідношень, а також що цілі класи алгебраїчних структур можуть стояти у взаємно однозначній відповідності з класами графів. Досліджено код Прюфера - спосіб компактного комп'ютерного зображення дерев, та запрограмовано перетворення кода Прюфера дерева та таблицею Келі T-групоїда, який цьому дереву відповідає. Ця робота має слугувати підгрунтям для подальшого дослідження алгебраїчних властивостей операцій, заданих на графах та орграфах. У подальшому планується введення нових бінарних операцій та дослідження гомоморфізмів між отриманими структурами.

Література

[1] L. Nebeský, A tree is a finite nonempty set with a binary operation, Mathematica Bohemica **125** (1998), 455–458.

[2] L. Nebeský, An algebraic characterization of geodetic graphs, Czechoslovak Mathematical Journal **123** (1995), 701–710.

[3] F. Harary, Graph theory. Addison-Wesley, Reading, Mass. 1969.

[4] B. Zelinka, The lattice of all subtrees of a tree, Mathematica Slovaca **27** (1977), 277–286.