ISSN 2707-1898 (print)

Український журнал інформаційних технологій
Ukrainian Journal of Information Technology
https://science.lpnu.ua/uk/ujit
https://doi.org/10.23939/ujit2021.03.037
Article received 17.03.2021 p.
Article accepted 01.06.2021 p.
UDK 004.02

Correspondence author
V. K. Ovsyak
ovsyak@rambler.ru

В. К. Овсяк[1], О. В. Овсяк[2], Ю. В. Петрушка[1]

[1] Українська академія друкарства, м. Львів, Україна
[2] Київський національний університет культури і мистецтв, м. Київ, Україна

ВПОРЯДКУВАННЯ ТА ВПОРЯДКОВУВАННЯ В ДИСКРЕТНІЙ МАТЕМАТИЦІ ТА ІНФОРМАТИЦІ

Досліджено наявні засоби впорядкувань й впорядковувань в деяких важливих розділах дискретної математики та інформатики, а саме: в теорії множин, класичній математичній логіці, теорії доведень (доказів), теорії графів, методі Поста, системі алгоритмічних алгебр, алгоритмічних мовах об'єктного і асемблерного програмування. Наведено декартів добуток множин, впорядковані пари і впорядковані *n*-ки, опис засобами теорії множин впорядкованої пари, які виконані Вінером, Хаусдорфом і Куратовським. Описано вимоги до відношень, якими впорядковуються множини. Важливість впорядкувань в класичній математичній логіці і теорії доведень проілюстровано прикладами обчислень значень істинності логічних формул і формальним виводом формули на підставі правил виводу і правил заміни. Впорядкування в теорії графів показано на прикладі блок-схеми алгоритму Евкліда, призначеного для знаходження найбільшого спільного дільника двох натуральних чисел. Описано впорядкування та впорядковування як настанов, утворених двома, трьома і чотирма впорядкованими полями так і наявне впорядкування настанов в програмі метода Поста. Показано, що програма, утворена пронумерованими настановами, з неповторюваними номерами настанов і наявністю єдиної настанови з номером 1.

Проілюстровано засоби системи алгоритмічних алгебр, які застосовуються для виконання впорядкувань й впорядковувань в теорії алгоритмів. Наведено операції системи алгоритмічних алгебр, які включають узагальнені на трьохзначний алфавіт операції Булевої алгебри та операторні операції операторної алгебри. Описано властивості операції композиції, яка призначена для опису впорядкувань операторів операторної алгебри системи алгоритмічних алгебр. Впорядкування виконувані засобами алгоритмічних мов програмування проілюстровано на гіпотетичному застосуванні сучасної мови об'єктного програмування C#. Програма має містити тільки один метод *Main*(), з якого починається виконання програми. Асемблерна програма мікропроцесора ARM має мати тільки одну директиву *ENTRY,* з якої починається виконання програми. Настанови впорядковуються послідовно зверху вниз у вигляді стовпця і записуються в оперативну пам'ять під послідовно впорядкованими адресами. Для виконання переходів застосовуються адреси, записані в настановах переходів. Вектор переривань містить фіксовані адреси комірок пам'яті, в яких записані початкові адреси програм опрацювання переривань.

Ключові слова: дискретна математика; алгебра; впорядкована пара; комп'ютерні науки; алгоритм; послідовність.

Вступ

Впорядкування застосовується як в області дискретної математики, так і прикладних областях, зокрема – в інформаційних технологіях та системах. Сучасна дискретна математика вважається теоретичною основою як інших розділів математики, зокрема математичної логіки, теорії доведень, теорії графів так і прикладних наук, до яких належить теорія алгоритмів, парадигми і методи програмування та синтезу і аналізу обчислювальної техніки інформаційних систем і технологій.

Для отримання впорядкувань, які є результатом виконання процесів впорядковувань, необхідні конкретні засоби. Зокрема для упорядкування елементів довільної множини необхідне відношення, яке є рефлексивне, транзитивне і антисиметричне. Здавалось би, що така методологія буде застосовуватись в усіх інших розділах дискретної математики та інформатики. Однак, уже в класичній математичній логіці, теорії доведень і теорії графів, а також в прикладних теоріях, зокрема, теорії алгоритмів застосовуються дещо інші засоби, які не мають властивостей рефлексивності, транзитивності та антисиметричності. Цим самим є актуальна проблема, як у створені загальних засобів впорядковувань, так і в окресленні адекватних застосувань дискретної математики і прикладних теорій, в процесах синтезу і аналізу впорядкованих моделей інформаційних технологій і інтелектуальних систем.

Актуальність адекватного опису впорядкувань і процесів впорядковувань, як в математиці так в інформатиці, обумовлена прямою залежністю результатів обчислень від коректного опису впорядкувань і процесів впорядковувань.

Об'єкт дослідження – процеси впорядковувань в дискретній математиці та інформаційних технологіях.

Предмет дослідження – методи і засоби впорядкувань й впорядковувань дискретної математики та інформатики.

Мета роботи – добір методів і засобів для адекватного опису моделей впорядкувань і процесів впорядковувань інформаційних технологій і систем.

Для досягнення зазначеної мети визначено такі основні *завдання дослідження*: методів опису впорядкувань й впорядковувань в теорії множин, математичній логіці, теорії доведень, теорії графів, теорії алгоритмів і програмуванні.

Наукова новизна отриманих результатів дослідження – вперше встановлено недостатності методів сучасної дискретної математики, системи алгоритмічних алгебр та її модифікації для адекватного аналітично-

го опису впорядкувань й впорядковувань в дискретній математиці та інформаційних технологіях.

Практична значущість результатів дослідження – можливості та обмеження методів впорядковувань й впорядкувань в моделюванні інформаційних технологій і інтелектуальних систем.

Аналіз останніх досліджень та публікацій. В теорії множин, яка є основою сучасної математики, поняття впорядкування виражене в декартовому добутку множин [4], [8], [11]. Декартовий добуток множин полягає в утворені упорядкованих елементів множини, яке є результатом множення множин. В загальному випадку, коли відбувається множення n множин, отримується упорядкована n-ка елементів. В частковому випадку, коли маємо декартовий добуток двох множин, мають місце упорядковані пари. У зв'язку з тим, що загально прийнято вважати теорію множин основою математики, то створені моделі опису впорядкувань множин засобами теорії множин.

Відомі моделі Вінера, Куратовського та інших авторів [5], [6], якими упорядковані пари описано засобами теорії множин. Для впорядковування множин застосовуються відношення з властивостями рефлексивності, транзитивності та антисиметричності [4], [8].

В класичній математичній логіці впорядкування та впорядковування логічних обчислень виконуються на підставі застосування дужок і загально прийнятого, так названого, "тіснішого" та "слабшого" зв'язування операцій [4], [8]. Теорія доведень для впорядкувань й впорядковувань застосовує правила виводу і правила заміни, які є тавтологіями [7], [8].

В теорії графів [2], [8] впорядкування та впорядковування виконуються на підставі декартового добутку множин з додатковими елементами у вигляді неорієнтованих чи орієнтованих в'язків між вершинами графів.

Теоретичною основою інформаційних технологій і систем є теорія алгоритмів [1], [8], [12]. Методи опису алгоритмів мають особливі засоби, які відмінні від класичних математичних засобів, опису впорядкувань й впорядковувань.

В об'єктному [9], [10] та асемблерному [3] програмуванні впорядкування та впорядковування основані на завданні початкової настанови і нумерації усіх інших настанов.

Результати дослідження та їх обговорення

Впорядкування та впорядковування в дискретній математиці. Найбільш широко в інформатиці взагалі та інформаційних технологіях зокрема застосовуються такі розділи дискретної математики, як теорія множин, математична логіка, теорія графів і теорія доведень.

Елементи впорядкування та впорядковування в теорії множин. *Впорядкування.* Відомо, що декартовим добутком [4], [11] $A_1 \times A_2 \times \ldots \times A_n$ множин $A_1, A_2, \ldots, A_n$ називається сукупність послідовностей (сукупність упорядкованих n-ок елементів) виду $(a_1, a_2, \ldots, a_n)$, де $a_i \in A_i$, $1 \le i \le n$.

У частковому випадку, коли $n = 2$, декартовий добуток

$$A \times B = \{(a,b) \mid a \in A, b \in B\} = C$$

є множиною C, утвореною упорядкованими парами (a, b).

Прийнято вважати [4], [8], що в упорядкованій (впорядкованій) парі (a, b) елемент a є першим, а елемент b – другим. Відповідно в упорядкованій n-ці елементів $(a_1, a_2, \ldots, a_n)$ елемент a_1 – перший, a_2 – другий, …, a_n – n-тий.

Приклад. Нехай $S = \{1, 2, 3, 4\}$ і $T = \{a, b, c\}$. Тоді декартовий добуток $S \times T$ є такою множиною упорядкованих пар

$$\begin{aligned} S \times T = \{&(1,a),(1,b),(1,c),\\ &(2,a),(2,b),(2,c),\\ &(3,a),(3,b),(3,c),\\ &(4,a),(4,b),(4,c)\}. \end{aligned}$$

Відомі [5], [6] описи впорядкованих пар з використанням множин:
множин Н. Вінера

$$(a,b) := \{\{\{a\}, \varnothing\}, \{\{b\}\}\};$$

множин Ф. Хаусдорфа

$$(a,b) := \{\{a, 1\}, \{b, 2\}\};$$

множин К. Куратовського впорядкована пара (a, b) множини $\{a, b\}$ описується як

$$(a,b) := \{\{a\}, \{a,b\}\},$$

де 1 і 2 – два різних об'єкти, відмінних від a і b.

Окрім вище наведених також відомі [6] описи упорядкованих пар Кантора – Фреге, Куайна – Россера і Морзе.

Впорядковування. Для здійснення впорядковування множин застосовують відношення часткового порядку, суворого порядку і відношення квазіпорядку [4], [8].

Бінарне відношення R, визначене на множині A, називається частковим порядком на A, якщо воно рефлексивне, транзитивне і антисиметричне, тобто якщо для довільних елементів a, b, c із A виконуються властивості:

- aRa (рефлексивність);
- aRb і bRc, то aRc (транзитивність);
- aRb і bRa, то $a = b$ (антисиметричність) [4], [8].

Частковий порядок на множині A, як правило позначають символом ≤, а саму частково упорядковану множину A називають скорочено *чум* і позначають $(A, \le)$ [4]. Окрім відношення часткового порядку ще застосовують відношення суворого порядку (<) і відношення квазіпорядку.

На частково упорядкованій множині $(A, \le)$ введені поняття мінімального (максимального), найбільшого (найменшого) елемента A, лінійного порядку на A (коли довільні два елементи із A порівнюються відносно ≤), лінійно упорядкованої множини (ланцюга) і добре упорядкованої множини (коли довільна її непуста підмножина має найменший елемент) [4], [8].

Відома [4] аксіома доброго упорядкування (довільну непусту множину можна добре упорядкувати).

Практичне застосування теоретикомножинної моделі впорядковування елементів a, b, c, …, t довільної скінченої множини $M = \{a, b, c, \ldots, t\}$ полягає на задані кожному елементу множини числа. В ідеальному випадку кожне приписане довільному елементу множини число має бути унікальним. Наприклад у вигляді нижніх індексів елементам a, b, c, …, t припишемо такі числа як 2, 0, 1, …, n. Внаслідок отримаємо множину з пронумерованими елементами $M^1 = \{a_2, b_0, c_1, \ldots, t_n\}$. Застосувавши відношення часткового порядку (≤) впоряд-

ковуємо елементи в порядку зростання номерів і отримуємо таке впорядкування елементів множини $M^1 = \{b_0, c_1, a_2, \ldots, t_n\}$.

Дещо інший і часто застосовуваний спосіб впорядковування полягає на використанні впорядкованості літер алфавіту. Обидва способи можуть бути застосовані для впорядковувань елементів скінчених множин з відносно невеликою кількістю впорядковуваних елементів. В інформаційних технологіях дані способи можуть бути застосовані для побудови моделей структур даних але їхніх засобів уже недостатньо для створення моделей функціонування інформаційних технологій і систем.

Впорядкування та впорядковування в класичній математичній логіці. Базовими операціями класичної математичної логіки є кон'юнкція (&), диз'юнкція $(\vee)$, інвертування $(\neg)$, імплікація $(\rightarrow)$ і кванторні операції загальності $(\forall)$ та існування $(\exists)$, які визначенні на множині логічних значень $\{0, 1\}$ [4], [8].

Застосування операцій кон'юнкції і диз'юнкції. Бінарні операції кон'юнкції і диз'юнкції є комутативними та асоціативними і тому не можуть застосовуватись для впорядкувань обчислень логічних значень формул. Але значення істинності логічних формул залежать від впорядкування обчислень логічних операцій. Нижче наведений приклад ілюструє це.

Приклад. Нехай є логічна формула $x \vee y \,\&\, z$. У табл. 1 наведено два варіанти обчислень її значень істинності.

Табл. 1. Можливі значення істинності формули $x \vee y \,\&\, z$

x	y	z	$x \vee y \,\&\, z$	$(x \vee y) \,\&\, z$
0	0	0	0	0
0	0	1	0	0
0	1	0	0	0
0	1	1	1	1
1	0	0	0	0
1	0	1	0	1
1	1	0	0	0
1	1	1	1	1

Значення істинності формули $x \vee y \,\&\, z$ виконано на підставі прийнятої умови, що операція кон'юнкції зв'язує тісніше (виконується першою) за операцію диз'юнкції. У формулі $(x \vee y) \,\&\, z$ для впорядкування обчислення значення істинності застосовано круглі дужки.

У цьому прикладі застосовано два варіанти впорядковування. Перший – за замовчуванням (прийнятою угодою старшинства операцій) і другий – застосуванням дужок. Приклад ілюструє залежність результату обчислень значень істинності від впорядкування операцій $(x \vee y \,\&\, z \neq (x \vee y) \,\&\, z)$.

Застосування операції імплікації. Операція імплікації не є ані комутативною ані асоціативною і це одна із причин її частого застосування, особливо в теорії доведень (доказів). Але значення істинності логічних формул теж залежить від впорядкувань імплікацій. Покажемо це на прикладі.

Приклад. Нехай маємо формулу $x \rightarrow y \rightarrow z$. В табл. 2 наведені значення істинності цієї формули для двох варіантів впорядкувань.

Як це показано у табл. 2, $(x \rightarrow (y \rightarrow z) \neq (x \rightarrow y) \rightarrow z)$ впорядкування імплікацій має істотне значення для обчислень значень істинності формул математичної логіки.

Табл. 2. Значення істинності імплікацій $x \rightarrow y \rightarrow z$ **і** $(x \rightarrow y) \rightarrow z$

x	y	z	$x \rightarrow (y \rightarrow z)$	$(x \rightarrow y) \rightarrow z$
0	0	0	1	0
0	0	1	1	1
0	1	0	1	0
0	1	1	1	1
1	0	0	1	1
1	0	1	1	1
1	1	0	0	0
1	1	1	1	1

Застосування кванторних операцій. Кванторні операції загальності $(\forall)$ та існування $(\exists)$ застосовуються для зв'язування предикатних змінних і в частковому випадку їх можна розглядати як узагальнення кон'юнкції та диз'юнкції [8].

Застосуванням кванторів $\forall$ і $\exists$ до бінарного предиката $P(x, y)$ утворюємо предикатні формули $\forall x \forall y P(x, y)$, $\forall y \forall x P(x, y)$, $\exists x \exists y P(x, y)$, $\exists y \exists x P(x, y)$, $\forall x \exists y P(x, y)$, $\exists x \forall y P(x, y)$, $\forall y \exists x P(x, y)$ і $\exists y \forall x P(x, y)$.

У предикатних формулах $\forall x \forall y P(x, y)$, $\forall y \forall x P(x, y)$, $\exists x \exists y P(x, y)$ і $\exists y \exists x P(x, y)$ впорядкування кванторів загальності та існування значення немає. Але впорядкування кванторів в формулах $\forall x \exists y P(x, y)$, $\exists x \forall y P(x, y)$, $\forall y \exists x P(x, y)$ і $\exists y \forall x P(x, y)$ має істотне значення.

Приклад. Нехай маємо предикат $<(x, y)$ (x менше y) і $x, y \in N$ (множина натуральних чисел). Тоді формула $\forall x \forall y <(x, y)$, як і формула $\forall y \forall x <(x, y)$, є хибна. Тобто формули $\forall x \forall y <(x, y)$ і $\forall y \forall x <(x, y)$ є рівноважними. Формули $\exists x \exists y <(x, y)$ і $\exists y \exists x <(x, y)$ є істинними і теж рівноважними. В такому разі впорядкування кванторів значення немає.

Тепер розгляньмо формули $\forall x \exists y <(x, y)$ і $\exists y \forall x <(x, y)$. Перша з них описує, що для кожного x існує y більше ніж x, що є істинним. Тоді як друга формула описує, що існує y таке, що усі x менші за y, що є хибним. Формули $\forall x \exists y <(x, y)$ і $\exists y \forall x <(x, y)$ не є рівноважними. Цим самим впорядкування кванторів має істотне значення.

Застосування дужок для впорядковування формул математичної логіки має істотний недолік, який полягає у тому, що дужки не наділені ніякими властивостями. Відсутність будь-яких властивостей не надає можливостей виконання перетворень над впорядкуваннями.

Довільне відношення може бути істинним або хибним і тому є предикатом з самого визначення відношення [4], [8]. Будь-яка функція або операція від n змінних описується предикатом від $n + 1$ змінної [4], [8]. У зв'язку з цим засоби математичної логіки можуть бути застосовані для побудови математичних моделей як структур даних так і процесів функціонування інформаційних технологій і систем. Однак відсутність будь-яких властивостей дужок, якими виконується впорядковування формул математичної логіки, не забезпечує можливості виконання перетворень математичних моделей інформаційних технологій і систем.

***Впорядкування в теорії доведень* (*доказів*).** Впорядкуванням в теорії доведень є формальний вивід, який утворений послідовністю формул, кожна з яких є або початковою формулою або отриманою з попередніх

формул застосуванням правил заміни чи правил виводу [7], [8].

Формальний вивід розгляньмо на прикладі виводу із початкових формул $A\,\&\,B$ і $A\vee B\to C$ формули C [8]. На підставі правила виводу, яке називається опусканням кон'юнкції $(A\,\&\,B\Rightarrow A)$, із формули $A\,\&\,B$ виводимо A. Застосувавши до A правило введення альтернативи $(A\Rightarrow A\vee B)$ виводимо $A\vee B$. До $A\vee B$ і $A\vee B\to C$ застосовуємо правило modus ponens $[(A\vee B)\ \&\ (A\vee B\to C)\Rightarrow C]$ внаслідок отримуємо формулу C. Даний вивід має такий вигляд: $A\,\&\,B$, A, $A\vee B$, $A\vee B\to C$, C.

У цьому виводі, наприклад, формула $A\vee B$ не може бути виведена, якщо попередньо не виведена формула A. У зв'язку з тим впорядкування формул виводу має істотне значення. Але формальний вивід має дещо інший вигляд. Суть у тому, що формула $(A\,\&\,B\Rightarrow A)$ і формула $(A\Rightarrow A\vee B)$ фактично пов'язані операцією кон'юнкції. Тому ці два кроки виводу описуються формулою $(A\,\&\,B\Rightarrow A)\,\&\,(A\Rightarrow A\vee B)$. Вона з правилом modus ponens $[(A\vee B)\ \&\ (A\vee B\to C)\Rightarrow C]$ утворює формулу

$$(A\,\&\,B\Rightarrow A)\ \&\ (A\Rightarrow A\vee B)\ \&$$
$$[(A\vee B)\ \&\ (A\vee B\to C)\Rightarrow C].$$

Власне вона описує формальний вивід C з формул $A\,\&\,B$ і $A\vee B\to C$. Але застосовувана між цими трьома кроками формального виводу операція кон'юнкції є комутативною та асоціативною. Тоді як другий крок виводу, отримання формули $A\vee B$, можливий тоді і тільки тоді, коли вже здійснений перший крок виводу, а власне попередньо виведена формула $(A\,\&\,B\Rightarrow A)$. Аналогічно третій крок виводу, отримання формули $[(A\vee B)\ \&\ (A\vee B\to C)\Rightarrow C]$, можливий тоді і тільки тоді, коли були виконані два попередні кроки виводу.

На підставі наведеного прикладу можна зробити висновок, що застосовувані в теорії доведень засоби впорядковувань виводу формул не є адекватними тому, що не враховують впорядкованість кроків формального виводу. Вони, як і засоби математичної логіки базуються на застосуванні дужок для виконання впорядковувань і тому їхнє застосування для математичного моделювання інформаційних технологій і систем є обмеженим.

Впорядкування в теорії графів. Графом G є впорядкована пара $G:=(V,E)$, де V – множина вершин (вузлів), а E – множина пар вершин з $V\times V$ (у випадку неорієнтованого графа – невпорядкованих), які називають ребрами [2], [8]. З визначення графа слідує, що впорядкування в графах базується на декартовому добутку множин (впорядкованих парах чи n-ах теорії множин).

Розгляньмо приклад упорядкованого графа. В області інформатики найбільше розповсюдження мають орієнтовані графи у вигляді блок-схем алгоритмів. У зв'язку з цим розглянемо блок-схему алгоритму Евкліда. Описує вона процес знаходження найбільшого спільного дільника двох натуральних чисел x і y таких, що $x>y$. Суть алгоритму полягає у знаходженні остачі від ділення x на y і порівнянні остачі з нулем. Блок-схема алгоритму наведена на рис. 1.

Ввід даних з клавіатури комп'ютера позначено символом $\leftarrow$, а його приписування змінній x – знаком =. У такому разі вираз $x=\leftarrow$ описує введення даних і приписування їх змінній x. Аналогічно описано введення і приписування другого числа $y=\leftarrow$.

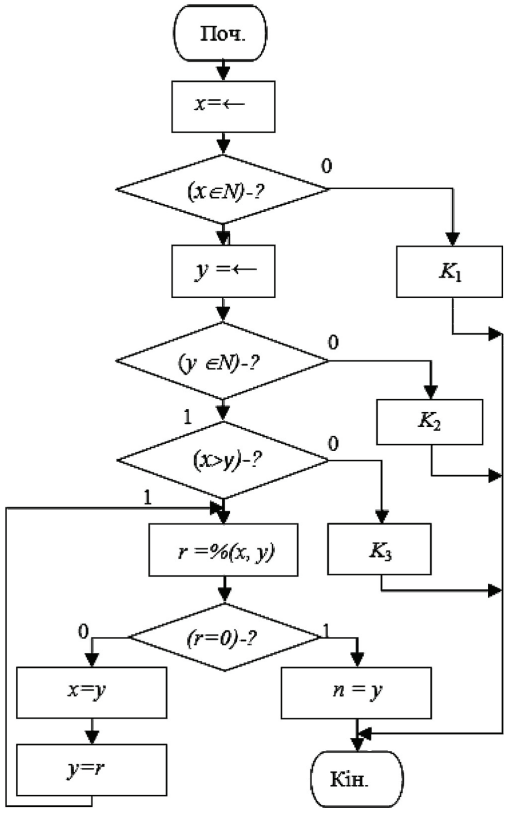


Рис. 1. Блок-схема алгоритму Евкліда

Перевірку належності $(\in)$ значень змінних x і y до натуральних чисел N записано як $(x\in N)-?$ і $(y\in N)-?$. Якщо ж значення змінних не належать до натуральних чисел, то на екран комп'ютера виводиться повідомлення K_1 і K_2, відповідно. K_3 – повідомлення про те, що натуральне число x не є більшим за натуральне число y. Знаходження остачі від ділення записано як $\%(x,y)$, а її приписування змінній $r-$ як $r=\%(x,y)$. Водночас n – змінна, якій приписується значення найбільшого спільного дільника.

В блок-схемі алгоритму рис. 1 вершинами є блок початку (Поч.), операторні (прямокутні) і умовні (у вигляді ромбів) блоки та блок кінця (Кін.) алгоритму.

Базуючись на впорядкованих n-ках і парах теорії множин можливо пронумерувати блоки, якими утворена блок-схема алгоритму. Після цього застосувати відношення часткового порядку $(\le)$ для впорядкування вершин графа (блок-схеми). Наприклад, з верху до низу блок-схеми блоки будуть пронумеровані числами 1, 2, 3, …, Але блок-схема алгоритму, яка подана на рис. 1, містить цикл. Він виражений переходом від операторного блоку з оператором $y=r$, до операторного блоку з оператором $r=\%(x,y)$. Нехай операторний блок з оператором $r=\%(x,y)$ отримав номер 9, а операторний блок з оператором $y=r$ – номер 13. В такому разі для переходу від блоку 13 до блоку 9 не буде виконуватися відношення часткового порядку $\le$.

Приклад ілюструє, що для впорядковувань вершин графів, частково блок-схем алгоритмів, є недостатньо впорядкованих n-нок і пар теорії множин. У зв'язку з цим впорядкування операторних і умовних блоків пред-

ставлено не тільки послідовністю їхнього розташування ай ще додатково орієнтованими зв'язками між ними.

Однак, орієнтовані зв'язки, як і дужки в математичній логіці і теорії доведень, не мають формальних властивостей, застосування яких надавало б можливості виконання перетворень моделей інформаційних технологій і систем. Тому застосування теорії графів, частково блок-схем алгоритмів, має істотне обмеження для моделювання інформаційних технологій і систем.

Впорядкування та впорядковування в інформатиці. Загально вважається, що теорія алгоритмів є теоретичною основою інформатики. Відома ціла низка методів опису алгоритмів. Зокрема це такі методи, як вербальний, блок-схем, машин Тюрінга, Поста, Колмогорова, з довільним доступом до пам'яті, Шонґаґе, а також рекурсивних функцій, нормальних алгорифмів Маркова, універсальних алгоритмів Крініцкого, числення λ, система алгоритмічних алгебр, модифікована система алгоритмічних алгебр, алгебра алгоритмів і модифікована алгебра алгоритмів. Серед них найбільш часто використовуваними є вербальний і блок-схемний методи.

Впорядкування та впорядковування в методі Поста. Серед "машинних" методів машина Поста [4] є методом найбільш наочним і доступним для сприйняття. Тому розгляньмо в чому полягає суть впорядкування та впорядковування в методі машини Поста.

Впорядкування в методі машини Поста виражене в форматах настанов, які є трьох типів. Перший тип має такі три поля:

"*Поле номера настанови*" "*Поле операції*" "*Поле номера наступної настанови*"

Другий формат настанови від першого формату відрізняється тим, що має два поля з номерами наступних настанов, а поле операції містить умовну операцію. Третій формат настанови, від двох попередніх, відрізняється тим, що має тільки поле номера настанови і поле операції, якою є операція зупинки функціонування машини.

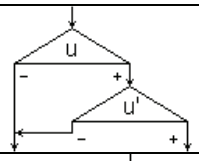
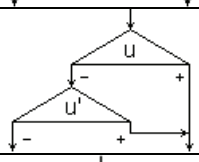
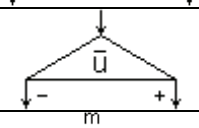
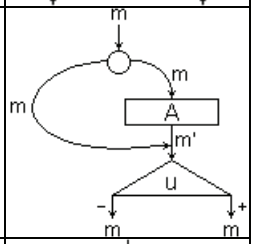
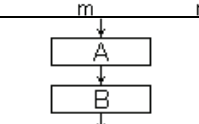
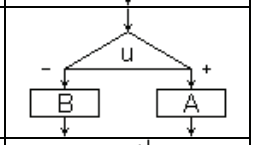
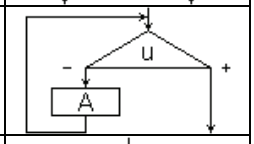
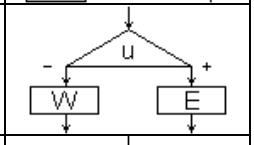
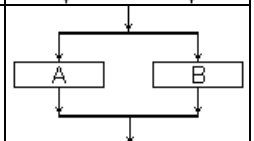
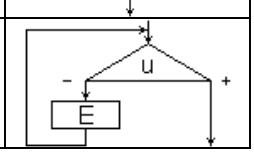
Впорядковування в методі машини Поста закладено в самій "програмі". Програмою машини Поста є послідовність настанов, яка має інструкцію в якій в "*Полі номера настанови*" є номер 1 та є наявні усі настанови з номерами в "*Полях номерів настанов*", які є в "*Полях номерів наступних настанов*".

Функціонування машини Поста *починається* з настанови, яка в "*Полі номера настанови*" має число 1. Після виконання операції, яка записана в "*Полі операції*", відбувається перехід до настанови, номер якої записаний у виконуваній настанови в "*Полі номера наступної настанови*".

Впорядковування, яке має місце в методі Поста, полягає в нумерації виконуваних настанов та задаванні номерів настанов, до яких здійснюється перехід після виконуваних настанов. Застосування такого методу для моделювання інформаційних технологій і систем забезпечує опис впорядкувань. Однак, над такими моделями інформаційних технологій і систем не можливо виконувати тотожні перетворення.

Впорядкування та впорядковування в системі алгоритмічних алгебр. Серед алгебричних методів опису алгоритмів найбільш відома система алгоритмічних алгебр В. Глушкова [1] та її модифікація Г. Цейтліна [12]. Модифікована система алгоритмічних алгебр утворена двома алгебрами. Розширеною на трьохзначний алфавіт (логічні значення 0, 1 і невизначене значення μ) алгеброю Буля та алгеброю операторів. Операції модифікованої системи алгоритмічних алгебр та їх інтерпретації наведено у табл. 3 [12].

Табл. 3. Операції модифікованої системи алгоритмічних алгебр

Тип	Назва операції	Форма запису: аналітична	природно-лінгвістична	графова
Логічні	Кон'юнкція	$u \,\&\, u'$ $u.u'$	*uIu'*	
	Диз'юнкція	$u \vee u'$ $u + u'$	*uАБОu'*	
	Заперечення	$\bar{u}$	$НЕ(u)$	
	Прогнозування (ліве множення умови на оператор)	$A \times u$	ПІСЛЯ *A* умова *u*	
Операторні	Композиція	$A * B$	*A* ПОТІМ *B*	
	Альтернатива	$([u]A,B)$	ЯКЩО *u* ТО *A* ІНАКШЕ *B*	
	Цикл	$\{[u]A\}$	ПОКИ НЕ *u* ЦИКЛ *A*	
	Фільтрація	$\underline{u}$	ФІЛЬТР (*u*)	
	Асинхронна диз'юнкція	$A \vee B$	*A* ПАРАЛЕЛЬНО *B*	
	Синхронізатор	$S(u)$	ЧЕКАТИ (*u*)	

В табл. 3 застосовані такі позначення: *u*, *u*' – логічні змінні, *A*, *B*, *E*, *W*– оператори, * – знак операції композиції, $\dot{\vee}$ – знак операції розпаралелення. Узагальнені логічні операції виконуються над змінними, які прий-

мають три значення, а саме 0, 1, μ [1]. Наприклад, у табл. 4 наведено визначення узагальненої кон'юнкції.

Табл. 4. Узагальнена кон'юнкція

$\alpha \ \& \beta$	0	1	μ
0	0	0	0
1	0	1	μ
μ	0	μ	μ

Операція композиції $P*Q$ (послідовне застосування операторів P і Q) асоціативна:

$$(P*Q)\ *R=P*\ (Q*R),$$

не є комутативна і одночасно для неї виконуються рівності:

$$P*E=E*P=P\ ,\ P*N=N*P=N\ ,$$

де E і N є сталі операторної алгебри [1].

Розглянемо приклад застосування модифікованої системи алгоритмічних алгебр. Наведений вище на рис. 1 алгоритм Евкліда є такою формулою модифікованої системи алгоритмічних алгебр:

$$x=\leftarrow\ *([x\in N]y=\leftarrow *([y\in N\,]\,([\,x>y\,]\ \{(r=\%(x,y))*$$
$$[r\neq 0]x=y*y=r\}*n=y,K_3),K_2),K_1).$$

Впорядкування та впорядковування в системі алгоритмічних алгебр та її модифікації, як це видно із наведеного прикладу, описано операцією композиції. Однак операція композиції є асоціативною. Наявність властивості асоціативності істотно обмежує застосування системи алгоритмічних алгебр та її модифікації. Вони можуть бути використані виключно для коректного опису неасоціативних моделей впорядкувань. Але, як правило, на практиці застосовуються власне неасоціативні впорядкування.

Впорядкування та впорядковування в алгоритмічних мовах програмування. Однією з алгоритмічних мов сучасного програмування є C# (сі шарп) [9], [10]. В програмі, яка написана мовою C# і має виконуватися, має бути метод, який має назву *Main*(). Власне з цього методу і починається виконання будь-якої програми, яка написана мовою C#. Поля, властивості, конструктори і методи мають знаходитись в класах.

Настанови, конструктори і методи коду програми, не тільки C#, а також усіх інших відомих мов об'єктного програмування і не тільки (а й інших парадигм програмування, зокрема, імперативної, процедурної, функціональної, логічної, декларативної, тощо і навіть найсучаснішої "хмарної"), розташовуються послідовно у вигляді стовпця або послідовно зліва на право. Настанови виконуються у порядку їхнього розташування з верху в низ або зліва на право. Методи виконуються у порядку їхнього розташування або вибором за їхніми назвами.

Для опису *статичних моделей* (самого опису складових, а не динаміки їхнього виконання) інформаційних технологій і систем цілком достатньо засобів теорії множин. Однак для створення *динамічних моделей* (опису процесів функціонування) інформаційних технологій і систем засобів впорядковувань теорії множин уже недостатньо, а наявні можливості мов будь-якої парадигми програмування не забезпечують засоби перетворень програм.

Впорядкування та впорядковування в мікропроцесорах. Ввімкнення мікропроцесора супроводжується формуванням сигналу обнулення, який формується і у випадку натиснення кнопки "*Reset*" на передній панелі комп'ютера. Після сигналу обнулення мікропроцесор видає на шину адрес, так названий, "початковий адрес". Як правило (залежить від фірми, яка виготовляє мікропроцесори), він має нульове значення або на декілька значень менше, ніж максимально допустимий розмір пам'яті.

Початковий адрес призначений для вибору з оперативної пам'яті першої настанови програми. Після чого мікропроцесором здійснюється дешифрування та виконання настанови. Як правило перша настанова містить адрес починаючи з якого в оперативній пам'яті розміщена програма, яка має виконуватись. Вибір із пам'яті наступних настанов відбувається на підставі автоматичного додавання до адресу виконуваної настанови числа, яке відповідає кількості байт виконуваної настанови. Винятки мають місце коли виконуються настанови переходу, скоку (*goto*) або переривань. Для опрацювання переривань застосовуються так названі "вектори переривань", які є набором фіксованих адрес комірок пам'яті. Вони призначені для зберігання перших настанов, які мають виконуватись у випадку появи переривань.

Програмування сучасними мовами *Асеблера* теж передбачає наявність директиви *ENTRY* [3], з якої починається виконання програми. Директива *ENTRY* має бути у виконуваному коді і не може бути більше однієї директиви *ENTRY* в одній програмі. Впорядкування блоків, настанов, макр і впорядкування їх виконання мало чим відрізняється від впорядкувань мовами об'єктного програмування.

Методи апаратурної реалізації мікропроцесорів, а також мови їхнього асемблерного рівня програмування, не надають можливостей перетворень впорядкувань.

Висновки

Впорядкування та впорядковування застосовується в дискретній математиці та інформатиці. Для впорядкування множин застосовуються відношення, які мають властивості рефлексивності (іррефлексивності) і транзитивності.

Впорядкування в класичній математичній логіці і теорії доведень базуються на загально прийнятих правилах обчислень значень істинності.

В теорії графів для впорядкувань вершин і зв'язків застосовують декартів добуток множин з додатковими елементами у вигляді неорієнтованих чи орієнтованих в'язків між вершинами графів та орієнтацію зв'язків.

В алгебричних теоріях алгоритмів для впорядковувань операторів введені спеціальні операції. Але наявність властивості асоціативності операції композиції не забезпечує однозначності впорядкувань операторів.

Наявні впорядкування в алгоритмічних мовах програмування не забезпечують можливостей перетворень таких впорядкувань, що було б доцільним з погляду підвищення швидкодії і оптимізації комп'ютерних систем. Повстала потреба введення позиційної логіки як теоретичної основи впорядкувань й впорядковувань в дискретній математиці та інформатиці.

Подяка. За підтримку низький уклін член-кореспонденту НАНУ, професору, д-ру технічних наук Володимиру Грицику і професору, д-ру фіз.-мат. наук Ярославу Драґану.

References

[1] Gluschkow, W. M., Zeitlin, G. E., & Justchenko, J. L. (1980). Algebra. Sprachen. Programmierung. Akademie-Verlag, Berlin.
[2] Graph (discrete mathematics). Retrieved from https://uk.wikipedia.org/wiki/Граф_(математика)
[3] Hohl, W. (2014). Asembler dla procesorów ARM. Gliwice: Helion.
[4] Kriviy, S.D. (2014). Discrete mathematics. Chernivtsi-Kyiv: Bookrek. [In Ukrainian].
[5] Kuratovsky, K., & Mostovsky, A. (1970). Set theory. Moscow: Mir, 360 p. [In Rusian].
[6] Ordered_pair. Retrieved from https://ru.other.wiki/wiki/Ordered_pair#Wiener%27s_definition
[7] Proof theory. Retrieved from https://uk.wikipedia.org/wiki/Теорія доведення
[8] Ross, K.A., & Wright, C.R.B. (2008). Matematyka dyskretna. Warszawa: Wydawnictwo naukowe PWN.
[9] Troelsen, E. (2011). C # 2010 programming language and .NET 4.0 platform. Moscow: OOO I. D. Williams. [In Rusian].
[10] Troelsen, E. (2007). C # 2005 programming language and .NET 2.0 platform. Moscow: OOO I. D. Williams. [In Rusian].
[11] Vinogradov, I. M. (Ed.). (1977–1985). *Mathematical encyclopedia* (Vol. 1–5). Moscow: Soviet Encyclopedia. [In Rusian].
[12] Zeitlin, G. E. (2003). Algebraic Algorithmics: Theory and Applications. *Cybernetics and Systems Analysis*, *39*(1), 6–15. [In Rusian]. https://doi.org/10.1023/A:1023860707232

V. K. Ovsyak[1], O. V. Ovsyak[2], J. V. Petruszka[1]

[1] *Ukrainian Academy of Printing, Lviv, Ukraine*
[2] *National University of Culture and Arts, Kyiv, Ukraine*

ORDER AND ORDERING IN DISCRETE MATHEMATICS AND INFORMATICS

The available means of ordering and sorting in some important sections of discrete mathematics and computer science are studied, namely: in the set theory, classical mathematical logic, proof theory, graph theory, POST method, system of algorithmic algebras, algorithmic languages of object-oriented and assembly programming. The Cartesian product of sets, ordered pairs and ordered *n*-s, the description by means of set theory of an ordered pair, which are performed by Wiener, Hausdorff and Kuratowski, are presented. The requirements as for the relations that order sets are described. The importance of ordering in classical mathematical logic and proof theory is illustrated by the examples of calculations of the truth values of logical formulas and formal derivation of a formula on the basis of inference rules and substitution rules. Ordering in graph theory is shown by the example of a block diagram of the Euclidean algorithm, designed to find the greatest common divisor of two natural numbers. The ordering and sorting of both the instructions formed by two, three and four ordered fields and the existing ordering of instructions in the program of Post method are described. It is shown that the program is formed by the numbered instructions with unique instruction numbers and the presence of the single instruction with number 1.

The means of the system of algorithmic algebras, which are used to perform the ordering and sorting in the algorithm theory, are illustrated. The operations of the system of algorithmic algebras are presented, which include Boolean algebra operations generalized to the three-digit alphabet and operator operations of operator algebra. The properties of the composition operation are described, which is intended to describe the orderings of the operators of the operator algebra in the system of algorithmic algebras. The orderings executed by means of algorithmic programming languages are demonstrated by the hypothetical application of the modern object-oriented programming language C#. The program must contain only one method *Main* () from which the program execution begins. The ARM microprocessor assembly program must have only one *ENTRY* directive from which the program execution begins.

Keywords: discrete mathematics; algebra; ordered pair; computer science; algorithm; sequence.

Інформація про авторів:

Овсяк Володимир Казимирович, д-р техн. наук, професор, кафедра автоматизації та комп'ютерних технологій. **Email:** ovsyak@rambler.ru; https://orcid.org/0000-0001-9295-284X

Овсяк Олександр Володимирович, д-р техн. наук, доцент, кафедра мистецтв. **Email:** ovsyak@ukr.net; https://orcid.org/0000-0003-2620-1938

Петрушка Юлія Володимирівна, магістр, кафедра автоматизації та комп'ютерних технологій. **Email:** julja-petrushka@rambler.ru

Цитування за ДСТУ: Овсяк В. К., Овсяк О. В., Петрушка Ю. В. Впорядкування та впорядковування в дискретній математиці та інформатиці. Український журнал інформаційних технологій. 2021, т. 3, № 1. С. 37–43.

Citation APA: Ovsyak, V. K., Ovsyak, O. V., & Petruszka, J. V. (2021). Order and ordering in discrete mathematics and informatics. *Ukrainian Journal of Information Technology*, *3*(1), 37–43. https://doi.org/10.23939/ujit2021.03.037