ФАКУЛЬТЕТ КОМП'ЮТЕРНИХ НАУК ТА КІБЕРНЕТИКИ
КИЇВСЬКОГО НАЦІОНАЛЬНОГО УНІВЕРСИТЕТУ
ІМЕНІ ТАРАСА ШЕВЧЕНКА

С.Л. Кривий
О.І. Ченцов

МАТЕМАТИЧНА ІНДУКЦІЯ
основи, застосування, моделі

Навчальне видання
(електронне видання)

Київ 2026

УДК 510.6.004.42
К82

У виданні розглядаються різні варіанти застосування методу математичної індукції в різних областях математичної науки. Зокрема, велика увага приділяється застосуванням цього методу у програмуванні та комп'ютерних науках. Метод математичної індукції працює на множинах, які повинні задовольняти певні умови. Ці умови випливають із самої природи методу і тісно пов'язані з порядком, яким упорядковуються елементи множини. Тому розглядаються різні відношення порядку, визначені на множинах де застосовний метод математичної індукції.

Застосування методу в області програмування дозволяє ввести поняття індуктивної множини та моделі, означення за індукцією й побудову за індукцією. Крім моделей індуктивного типу розглядаються логічні моделі, орієнтовані на конкретні предметні області. До таких предметних областей відноситься індуктивний спосіб побудови мов та структур даних.

Для студентів, що навчаються за освітніми програмами магістерського рівня за спеціальністю "Програмне забезпечення систем" та бакалаврського рівня "Програмна інженерія" за спеціальністю F2 "Інженерія програмного забезпечення", а також інших викладачів, аспірантів і студентів, зацікавлених у питаннях описаних вище.

Рецензенти:

Заславський В. А., доктор технічних наук, професор, Київський національний університет імені Тараса Шевченка.
Шевченко В. П., кандидат фізико-математичних наук, доцент, Київський національний університет імені Тараса Шевченка.

Рекомендовано до друку вченою радою факультету комп'ютерних наук та кібернетики (протокол № 13 від 21 квітня 2026 року)

Ухвалено науково-методичною комісією факультету комп'ютерних наук та кібернетики (протокол № 10 від 20 квітня 2026 року)

Зміст

ПЕРЕДМОВА

Inductio — латинське слово, що означає “наведення”. Стосовно математичної науки це слово означає метод, за допомогою якого можна перейти від окремого випадку справедливості деякого твердження до загального випадку його справедливості. Метод математичної індукції це один з важливих методів доведення математичних тверджень, який остаточно закріпився в математиці в XVII-му столітті, а вперше він був застосований у 1575 році італійським вченим Ф. Мауроліко. Остаточне його становлення пов’язують з іменами П. Ферма (який називав його методом нескінченного спуску), Б. Паскаля та Я. Бернуллі. Сам вираз “математична індукція” ввів А. де Морган на початку XIX-го століття. Тепер цей метод описаний у багатьох джерелах, як наукового, так і філософського характеру.

Метод математичної індукції, будучи за своєю природою пов’язаним з поняттям натурального числа, застосовний передусім у дискретних областях сучасної математики. Ось деякі з таких областей.

Теорія чисел: доведення тверджень теорії чисел, тотожностей, рівностей та нерівностей.

Комбінаторика: встановлення формул комбінаторних обрахунків (кількість підмножин n-елементної множини, кількість відношень, кількість перестановок з повтореннями і без повторень тощо).

Алгебра: доведення алгебричних тверджень, тотожностей, нерівностей.

Тригонометрія: доведення тригонометричних тотожностей та тверджень, які важливі в математичному аналізі та побудові ефективних програм у програмуванні.

Геометрія: доведення властивостей геометричних фігур.

Теорія графів: доведення структурних властивостей графів, зокрема, властивості ойлерових та гамільтонових графів.

Програмування: доведення правильності програм, доведення термінальності в алгоритмах і програмах.

Теоретичні основи інформатики: індуктивні множини, означення за індукцією, побудова за індукцією тощо.

Деякі важливі твердження, тотожності, рівності та нерівності, які доводяться методом математичної індукції, винесені у завдання для самостійної роботи.

Основною метою видання є допомогти студентам і аспірантам освоїти метод математичної індукції та області його застосування.

Автори

Розділ 1

ЗАСТОСУВАННЯ МЕТОДУ МАТЕМАТИЧНОЇ ІНДУКЦІЇ

Якщо аналізувати метод індукції, то чітко вимальовуються три особливості цього методу: *узагальнення, спеціалізація та аналогія* [1].

Для пояснення цих особливостей розглянемо яку-небудь універсальну множину U і нехай $A \subseteq U$ – деяка її підмножина.

Узагальненням називається перехід від розгляду елементів множини A до розгляду елементів множини $B \subseteq U$, яка є надмножиною множини A, тобто $A \subset B$. Наприклад, ми виконуємо узагальнення, коли переходимо від розгляду трикутників до розгляду многокутників з довільним числом сторін, від лінійних рівнянь до рівнянь вищих степенів тощо.

Під час узагальнення часто виконується перехід від одного лише елемента до цілого класу, якому належить цей елемент.

Спеціалізацією називається перехід від розгляду елементів множини B до розгляду елементів її підмножини A, тобто $A \subset B$. Наприклад, ми виконуємо перехід від системи рівнянь степеня ≥ 2 до розгляду системи лінійних рівнянь або перехід від розгляду многокутників до розгляду трикутників.

Під час спеціалізації, на відміну від узагальнення, часто виконується перехід від класу елементів до конкретного елемента, який належить цьому класу.

Аналогія, на жаль, не є особливістю, так добре зрозумілою, як узагальнення та спеціалізація, і потребує певної обережності у використанні. Проводячи аналіз проблеми за аналогією доводиться звертатися до аналогічних проблем чи явищ, які розглядалися в минулому і про які вже дещо відомо. Подібність явищ, які полегшують розв'язання проблеми, може виявитися настільки значною, що виникає бажання розповсюдити міркування і висновки так, що вони можуть вийти за межі реальності. Дійсно, значна подібність проблем робить нас менш обережними і часто призводить до неправильних висновків. Отже, для ефективного використання аналогії слід бути обачним і виконувати певні умови. Ці умови, як правило, зводяться до таких рекомендацій:

а) спершу слід детально вивчити проблему для того, щоб можна було чітко визначити елементи, за якими встановлюватиметься аналогія;

б) проаналізувати відомі проблеми та методи їх розв'язання, які аналогічні проблемі, що вивчається;

в) вивчити невідому проблему, порівнюючи її з відомими аналогічними проблемами. При цьому важливо встановити як риси подібності, так і риси розбіжності, все це зважити і оцінити. З метою досягнення більшої подібності в аналогіях корисно виявляти, перш за все, відмінності між ними, а не подібності. Це дає змогу точніше описати властивості елементів у проблемах, за якими вони вважатимуться аналогічними.

Отже, аналогія – це певного типу подібність. В аналогії суттєвою відмінністю від інших типів подібності є те, що подібні елементи знаходяться в деякому відношенні. Два елементи $a \in A$ і $b \in B$ вважаються аналогічними, якщо $(a, b) \in R \subseteq A \times B$, де R – відношення подібності (це відношення часто є рефлексивним і симетричним).

Але так визначене відношення подібності теж не є досить зрозумілим. Для більшої конкретизації цього відношення вводиться поняття *абстракції*. Коли в процесі пошуку аналогії виділяємо певні властивості елементів із множин A і B та абстрагуємося від інших, несуттєвих з нашого погляду, то елементи $a \in A$ і $b \in B$ вважаються аналогічними, якщо вони мають однакові виділені властивості.

Зрозуміло, що відношення подібності може бути не єдиним. Таких відношень може існувати декілька, що означає, що між елементами множин A і B повинно існувати декілька аналогій.

Узагальнення, спеціалізація та аналогія часто співпрацюють при розв'язуванні математичних задач. Для ілюстрації цієї співпраці в виданні наводиться велика кількість прикладів з різних областей математики і, зокрема, з області науки про обчислення.

Метод математичної індукції є потужним засобом, який дозволяє працювати з нескінченними множинами і потребує лише двох етапів – перевірити *базу індукції* та обґрунтувати *індукційний перехід*.

Позначимо $\mathcal{N} = \{0, 1, 2, 3, \dots\}$ множину натуральних чисел і $\mathcal{N}^+ = \{1, 2, 3, \dots\}$ — множину додатних натуральних чисел.

Класичний метод математичної індукції визначається на множині натуральних чисел $\mathcal{N}$ і формулюється у такому вигляді.

Теорема 1.1 (метод математичної індукції). *Твердження $P(n)$ справедливе для довільного натурального числа n, якщо*

1) P справедливе для $n = 0$ і

2) із справедливості P для деякого $n = k > 0$ випливає справедливість P і для n=k+1.

Доведення. Припустимо що твердження теореми хибне, тобто воно справедливе не для довільного натурального числа n. Тоді існує таке натуральне число m, що твердження теореми хибне для $n = m$ і для довільного $n < m$ твердження справедливе.

Отже, m – це перше натуральне число, для якого твердження теореми хибне. Очевидно, що $m > 0$, оскільки для $n = 0$ твердження справедливе (умова 1). Тоді для натурального числа $m - 1$ твердження справедливе, а для наступного натурального числа m твердження хибне. Але цей висновок суперечить умові 2). ■

Умови 1) і 2) мають особливе значення. Умова 1) створює **базу** для ведення індукції, а умова 2) дає право **поширити цю базу**, тобто право переходу від частинного випадку до загального.

Для ілюстрації роботи методу математичної індукції розглянемо його застосування у згаданих вище областях.

1.1. Індукція у теорії чисел

Множина натуральних чисел є основою теорії чисел, де принцип математичної індукції успішно працює.

Задача 1.1.1. Довести рівність $S_n = 1 + 3 + 5 + \ldots + 2n - 1 = n^2$.

Доведення. Умова 1) справджується для $n = 1$, оскільки $1 = 1^2$.

(Умова 2) Припустимо, що рівність справджується для $n = k$, тобто $S_k = k^2$. Тоді

$$S_{k+1} = S_k + (2(k+1) - 1) = k^2 + 2k + 1 = (k+1)^2.$$

Рівність доведено.

Задача 1.1.2. Довести рівність

$$S_n = 1^2 - 2^2 + 3^2 - 4^2 + \ldots (-1)^{n-1} n^2 = (-1)^{n-1} \frac{n(n+1)}{2}.$$

Доведення. Умова 1) для $n = 1$ справджується, оскільки $1^2 = (-1)^0 = 1$.
(Умова 2) Припустимо, що для $n = k$ $S_k = (-1)^{k-1} \frac{k(k+1)}{2}$. Тоді

$$\begin{aligned} S_{k+1} = S_k + (-1)^k (k+1)^2 = (-1)^{k-1} \tfrac{k(k+1)}{2} + (-1)^k (k+1)^2 = \\ = (-1)^k [(k+1) - \tfrac{k}{2}](k+1) = (-1)^k \tfrac{(k+1)(k+2)}{2}. \end{aligned}$$

Рівність доведено.

Задача 1.1.3. Довести рівність

$$1 \cdot 2 + 2 \cdot 3 + 3 \cdot 4 + \ldots + (n-1)n = \frac{(n-1)n(n+1)}{3}.$$

Доведення. (Умова 1) для $n = 2$ справджується, оскільки $1 \cdot 2 = \frac{1 \cdot 2 \cdot 3}{3}$.
(Умова 2) Припустимо, що $n = k$ справджується $S_k = \frac{(k-1)k(k+1)}{3}$. Тоді

$$S_{k+1} = S_k + k(k+1) = \frac{(k-1)k(k+1)}{3} + k(k+1) = \frac{k(k+1)(k+2)}{3}.$$

Рівність доведено.

Задача 1.1.4. Довести рівність

$$1 \cdot 2 \cdot 3 + 2 \cdot 3 \cdot 4 + \ldots + n(n+1)(n+2) = \frac{n(n+1)(n+2)(n+3)}{4}.$$

Доведення. (Умова 1) для $n = 3$ справджується, оскільки $1 \cdot 2 \cdot 3 = \frac{1 \cdot 2 \cdot 3 \cdot 4}{4}$.
(Умова 2) Припустимо, що для $n = k$ рівність $S_k = \frac{k(k+1)(k+2)(k+3)}{4}$ справджується. Тоді

$$S_{k+1} = S_k + (k+1)(n+2)(n+3) =$$
$$= \frac{k(k+1)(k+2)(k+3)}{4} + (k+1)(k+2)(k+3) =$$
$$= (k+1)(k+2)(k+3)\left(\frac{k}{4} + 1\right) = \frac{(k+1)(k+2)(k+3)(k+4)}{4}.$$

Рівність доведено.

Задача 1.1.5. Довести рівність

$$\frac{1}{1 \cdot 3} + \frac{1}{3 \cdot 5} + \ldots + \frac{1}{(2n-1)(2n+1)} = \frac{n}{2n+1}.$$

Доведення. (Умова 1) для $n = 1$ справджується, оскільки $\frac{1}{1 \cdot 3} = \frac{1}{3}$.
(Умова 2) Припустимо, що для $n = k$ рівність $S_k = \frac{k}{2k+1}$ виконується. Тоді

$$S_{k+1} = S_k + \frac{1}{(2k+1)(2k+3)} = \frac{k}{2k+1} + \frac{1}{(2k+1)(2k+3)} =$$
$$= \frac{1}{2k+1}\left(k + \frac{1}{2k+3}\right) = \frac{1}{2k+1}\frac{2k^2+3k+1}{2k+3} = \frac{1}{2k+1}\frac{(k+1)(2k+1)}{2k+3} =$$
$$= \frac{k+1}{2k+3} = \frac{k+1}{2(k+1)+1}.$$

Рівність доведено.

Задача 1.1.6. Довести рівність

$$S_n = \frac{1}{1\cdot 5} + \frac{1}{5\cdot 9} + \frac{1}{9\cdot 13} + \ldots + \frac{1}{(4n-3)(4n+1)} = \frac{n}{4n+1}.$$

Доведення. (Умова 1) справедлива для $n = 1$, оскільки $\frac{1}{1\cdot 5} = \frac{1}{5}$.

(Умова 2) Припустимо, що для $n = k$ рівність $S_k = \frac{k}{4k+1}$ виконується. Тоді

$$S_{k+1} = S_k + \frac{1}{(4k+1)(4k+5)} = \frac{k}{4k+1} + \frac{1}{(4k+1)(4k+5)} =$$

$$= \frac{1}{4k+1}\left(k + \frac{1}{4k+5}\right) = \frac{1}{4k+1}\left(\frac{4k^2+5k+1}{4k+5}\right) = \frac{k+1}{4k+5},$$

оскільки $4k^2 + 5k + 1 = (k+1)(4k+1)$.

Рівність доведено.

Задача 1.1.7. Довести, що коли $u_0 = 2, u_1 = 3$ і для довільного натурального числа k, де $k \geq 1$, справедлива рівність

$$u_{k+1} = 3u_k - 2u_{k-1} \text{ то } u_k = 2^k + 1.$$

Доведення. (Умова 1) справедлива для $k = 0$ і $k = 1$.

(Умова 2) Припустимо, що

$$u_{k-1} = 2^{k-1} + 1, \quad u_k = 2^k + 1.$$

Тоді

$$u_{k+1} = 3(2^k + 1) - 2(2^{k-1} + 1) = 2^{k+1} + 1.$$

Рівність доведено.

Розглянуті задачі були простими в тому сенсі, що в рівностях, які в них доводилися, було явно вказано ліві і праві частини. Але коли задано тільки одну зі сторін, то другу сторону потрібно знайти. Тут необхідно побудувати кілька окремих значень і помітити в них закономірність. Ця закономірність стає гіпотезою, правильність якої перевіряють методом математичної індукції.

Задача 1.1.8. Обчислити суму членів послідовності

$$S_n = 1\cdot 1! + 2\cdot 2! + 3\cdot 3! + \ldots + n\cdot n!.$$

Розв'язання. Обчислимо частинні суми

$$S_1 = 1\cdot 1! = 1, \quad S_2 = 1\cdot 1! + 2\cdot 2! = 5,$$

$$S_3 = 1 \cdot 1! + 2 \cdot 2! + 3 \cdot 3! = 23, \quad S_4 = 1 \cdot 1! + 2 \cdot 2! + 3 \cdot 3! + 4 \cdot 4! = 119.$$

Якщо проаналізувати отримані значення, то проглядається така закономірність:

$$S_1 = 2! - 1, \; S_2 = 3! - 1, \; S_3 = 4! - 1, \; S_4 = 5! - 1.$$

Це є підставою для формулювання припущення, що $S_n = (n+1)! - 1$.

Перевіримо справедливість нашого припущення методом математичної індукції.

(Умова 1) справедлива, оскільки $S_1 = 1 \cdot 1! = 2! - 1$.

(Умова 2) Припустимо, що для $n = k$

$$S_k = 1 \cdot 1! + 2 \cdot 2! + 3 \cdot 3! + \ldots + k \cdot k! = (k+1)! - 1.$$

Тоді

$$S_{k+1} = S_k + (k+1)(k+1)! =$$
$$= (k+1)! - 1 + (k+1)(k+1)! = (k+1)!(1 + k + 1) - 1 = (k+2)! - 1.$$

Задача 1.1.9. Довести, що для цілого $n \geq 0$ число

$$A_n = 11^{n+2} + 12^{2n+1}$$

ділиться на 133.

Розв'язання. Умова 1) для $n = 0$ виконується, оскільки $11^2 + 12 = 133$.

(Умова 2) Припустимо, що A_n справедливе для $n = k$. Тоді

$$A_{k+1} = 11^{k+3} + 12^{2k+3} = 11 \cdot 11^{k+2} + 144 \cdot 12^{2k+1} =$$
$$= 11 \cdot 11^{k+2} + 133 \cdot 12^{2k+1} + 11 \cdot 12^{2k+1} =$$
$$= 11(11^{k+2} + 12^{2k+1}) + 133 \cdot 12^{2k+1} = 11A_k + 133 \cdot 12^{2k+1}.$$

На підставі припущення індукції кожний з доданків ділиться на 133, а тоді і їхня сума ділиться на 133.

Задача 1.1.10. Довести, що довільну суму грошей, більшу 6 гривень, можна розміняти купюрами вартістю 2 і 5 гривень.

Доведення. Умова 1) справджується, оскільки для $n = 7$ маємо $7 = 2+5$.

(Умова 2) Припустимо, що твердження справедливе для $n > 7$. Тоді можливі два випадки: а) n розмінюється лише купюрами 2 гривні і б) n розмінюється купюрами, серед яких є хоча б одна купюра вартістю 5 гривень

У випадку а) купюр повинно бути не менше чотирьох, оскільки $n > 7$. Для того щоб розміняти $n + 1$ гривень, потрібно замінити дві купюри вартістю 2 гривні однією купюрою вартістю 5 гривень.

У випадку б) купюру вартістю 5 гривень потрібно замінити трьома купюрами вартістю 2 гривні.

Розв'язуваність цієї задачі в загальному випадку випливає з відомої властивості найбільшого спільного дільника цілих чисел: для довільних цілих чисел a, b існують такі цілі числа x, y, що НСД(a, b) $= n = ax + by$. У випадку взаємно простих чисел a, b їхній НСД дорівнює 1. А звідси випливає, що довільну суму n гривень можна розміняти купюрами вартістю a і b гривень, оскільки з них можна одержати вартість 1 гривня. Вартості купюр 2 і 5 є числами взаємно простими.

Задача 1.1.11. Довести, що сума кубів трьох послідовних натуральних чисел ділиться на 9.

Доведення. Умова 1) справджується, оскільки для $1^3 + 2^3 + 3^3 = 36$ ділиться на 9 коли першим натуральним числом є 1.

(Умова 2) Припустимо, що твердження справедливе для $n = k$,тобто

$$k^3 + (k+1)^3 + (k+2)^3$$

ділиться на 9. Тоді

$$(k+1)^3 + (k+2)^3 + (k+3)^3 = (k+1)^3 + (k+2)^3 + k^3 + 9k^2 + 27k + 27 = \\ = [k^3 + (k+1)^3 + (k+2)^3] + 9(k^2 + 3k + 3).$$

На підставі припущення індукції кожний з доданків ділиться на 9, тоді і сума цих доданків ділиться на 9.

Задача 1.1.12. Довести рівність (біном Ньютона)

$$(a+b)^n = C_n^0 a^n b^0 + C_n^1 a^{n-1} b^1 + C_n^2 a^{n-2} b^2 + \ldots + C_n^n a^0 b^n. \tag{1.1}$$

Доведення Умова 1) справджується, оскільки для $n = 1$ маємо

$$(a+b)^1 = C_1^0 a^1 b^0 + C_1^0 a^0 b^1 = a + b.$$

(Умова 2) Нехай теорема справедлива для $n = k$,тобто

$$(a+b)^k = C_k^0 a^k b^0 + C_k^1 a^{k-1} b^1 + C_k^2 a^{k-2} b^2 + \ldots + C_k^k a^0 b^k.$$

Тоді для $n = k + 1$ за припущенням індукції маємо

$$(a+b)^{k+1} = (a+b)^k (a+b) = (C_k^0 a^k b^0 + C_k^1 a^{k-1} b + \ldots C_k^k a^0 b^k)(a+b) = \\ = C_k^0 a^{k+1} b^0 + (C_k^1 a^k b + C_k^0 a^k b) + (C_k^2 a^{k-1} b^2 + C_k^1 a^{k-1} b^2) + \ldots + C_k^k a^0 b^{k+1} = \\ = C_{k+1}^0 a^{k+1} b^0 + C_{k+1}^1 a^k b + \ldots + C_{k+1}^k a^1 b^k + C_{k+1}^{k+1} a^0 b^{k+1},$$

оскільки $C_n^m + C_n^{m-1} = C_{n+1}^m$, а $C_n^0 = C_{n+1}^0 = 1$.

Задача 1.1.13. Довести, що $n^3 - n$ ділиться на 3.

Доведення. Умова 1) справджується для $n = 1$, оскільки $1^3 - 1 = 0$, а 0 ділиться на 3.

(Умова 2) Нехай твердження справджується для $n = k$. Тоді

$$(k+1)^3 - (k+1) = k^3 + 3k^2 + 3k + 1 - k - 1 = (k^3 - k) + 3k(k+1).$$

Обидва доданки діляться на 3, тоді і їхня сума теж ділиться на 3.

Це твердження можна довести без застосування методу математичної індукції. Справді, $n^3 - n = n(n^2 - 1) = (n-1)n(n+1)$, тоді серед трьох послідовних натуральних чисел є число, кратне 3.

Задача 1.1.14. Довести, що $n^5 - n$ ділиться на 5.

Доведення. Умова 1) справджується для $n = 1$, оскільки $1^5 - 1 = 0$, а 0 ділиться на 5.

(Умова 2) Нехай твердження справджується для $n = k$. Тоді на підставі біному Ньютона отримуємо

$$(k+1)^5 - (k+1) = k^5 + 5k^4 + 10k^3 + 10k^2 + 5k + 1 - k - 1 =$$

$$= (k^5 - k) + 5k(k^4 + 2k^3 + 2k^2 + 1).$$

Обидва доданки діляться на 5, тоді і їхня сума теж ділиться на 5.

З задачами 1.1.13 і 1.1.14 пов'язана така історія. Знаменитий німецький математик XVII-го століття Г. В. Лейбніц довів справедливість цих задач і справедливість того, що $n^7 - n$ кратне 7 (див. Задачі для самостійної роботи). На цій підставі він припустив, що $n^k - n$ ділиться на k для довільного непарного k і довільного $n \in \mathcal{N}^+$. Але швидко сам побачив хибність свого припущення, знайшовши що $2^9 - 1 = 510$ не ділиться на 9.

Задача 1.1.15. Послідовність чисел Фібоначчі задається такими співвідношенням:

$$u_n = \begin{cases} 1, & \text{якщо} \quad n = 1; \\ 1, & \text{якщо} \quad n = 2; \\ u_{n-1} + u_{n-2}, & \text{якщо} \quad n > 2. \end{cases}$$

Довести, що число Фібоначчі кратне 3, якщо його номер кратний 4.

Доведення. Умова 1) виконується, оскільки $u_4 = 3$ кратне 3.

(Умова 2) Припустимо: що u_{4k} кратне 3 для $n = 4k$. Тоді

$$u_{4(k+1)} = u_{4k+3} + u_{4k+2} = u_{4k+2} + u_{4k+1} + u_{4k+2} =$$

$$= u_{4k+1} + u_{4k} + u_{4k+1} + u_{4k+1} + u_{4k} = 2u_{4k} + 3u_{4k+1}.$$

На підставі припущення індукції обидва доданки кратні 3, а тоді і їхня сума кратна 3. А це і потрібно було довести.

Задача 1.1.16. Довести основну теорему арифметики [3]: "*Кожне натуральне число $n > 1$ можна представити у вигляді добутку простих чисел*".

Доведення. Нехай $P(n)$ означає "n є добутком простих чисел". Найменшим числом в цьому випадку є число 2.

(Умова 1) Оскільки 2 є простим числом, то очевидно, що $P(2)$ справджується.

(Умова 2) Припустимо, що $n > 2$ і $P(k)$ має місце для всіх $2 \leq k < n$. Можливі два випадки: а) число n просте і б) число n складене.

У випадку а) все доведено, а у випадку б) отримуємо $n = x \cdot y$, де $x, y \in \mathcal{N}^+$ і $2 \leq x, y \leq n$. На підставі припущення математичної індукції отримуємо справедливість $P(x)$ і $P(y)$.[1] Звідси випливає, що x і y є добутками простих чисел, але це означає, що $n = x \cdot y$ теж є добутком простих чисел.

З наведених прикладів випливає, що методом математичної індукції можна доводити і такі твердження, які для $n = 0$ і для $n = 1$ не справджуються, але які справджуються для деякого натурального числа $n > 1$. Наприклад, довести, що $2^n > n^2$. Це твердження хибне для $n = 1, 2, 3, 4$ і справджується для першого $n = 5$. Тут перевірку виконання індукційних умов 1) і 2) слід починати з $n = 5$. Звідси випливає такий висновок: *твердження може справджуватися у великій кількості окремих випадків, але не справджуватися в загальному випадку.*

На підтвердження цього висновку наведемо відомий приклад з теорії чисел, де потрібно перевірити що число $991n^2 + 1$ не є повним квадратом для довільного $n \in \mathcal{N}^+$. Підставляючи у вираз $991n^2 + 1$ послідовно значення $n = 1, 2, 3, 4, 5, \ldots$ ми за життя ніколи не отримаємо повний квадрат. Насправді число $991n^2 + 1$ буде повним квадратом для багатьох натуральних чисел n, але першим таким числом у цій послідовності є число

$$n = 12055735790331359447442538767.$$

ЗАДАЧІ ДЛЯ САМОСТІЙНОЇ РОБОТИ

1. Довести такі твердження [7]:

а) $n^7 - n$ кратне 7;

б) $n(2n^2 - 3n + 1)$ кратне 6; $5 \cdot 2^{3n-2} + 3^{3n-1}$ кратне 19;

в) $1^2 + 2^2 + \ldots + n^2 = \dfrac{n(n+1)(2n+1)}{6}$;

г) $1^3 + 2^3 + \ldots + n^3 = \left[\dfrac{n(n+1)}{2}\right]^2$;

д) $1^2 + 3^2 + \ldots + (2n-1)^2 = \dfrac{n(4n^2-1)}{3}$;

е) $\dfrac{1}{1 \cdot 2} + \dfrac{1}{2 \cdot 3} + \ldots + \dfrac{1}{n \cdot (n+1)} = \dfrac{n}{n+1}$;

[1] тут використаний метод *повної* математичної індукції

є) $(1-1/4)\cdot(1-1/9)\cdot\ldots\cdot(1-1/(n+1)^2)=(n+2)/(2n+2)$;
ж) коли $n\geq 2$, то $1/(n+1)+1/(n+2)+\ldots+1/2n>13/24$;
з) $1\cdot 2+2\cdot 5+\ldots+n\cdot(3n-1)=n^2(n+1)$;
і) $1/2\cdot 3/4\cdot 5/6\cdot\ldots\cdot(2n-1)/2n\leq\frac{1}{\sqrt{3n+1}}$;
к) коли $n\geq 2$, то число $2^{2^n}+1$ закінчується числом 7;

2. Довести, що для всіх $n\geq 3$ справедлива нерівність $2^n>2n+1$.
3. Довести, що число Фібоначчі
   а) парне, коли його номер кратний 3;
   б) кратне 4, коли його номер кратний 6;
   в) кратне 5, коли його номер кратний 5.
4. Довести, що коли $x_1,x_2,\ldots,x_n$ – додатні дійсні числа, такі що

$$x_1\cdot x_2\cdot\ldots\cdot x_n=1,\quad\text{то}\quad x_1+x_2+\ldots+x_n\geq n.$$

5. Довести, що коли $x_1,x_2,\ldots,x_n$ додатні дійсні числа такі, що

$$x_1+x_2+\ldots+x_n\leq 1/2,\ \text{то}\ (1-x_1)\cdot(1-x_2)\cdot\ldots\cdot(1-x_n)\geq 1/2.$$

1.2. Індукція у тригонометрії

Тригонометричні тотожності відіграють важливу роль не тільки у тригонометрії, а і у математичному аналізі та геометрії. Зокрема у математичному аналізі такі тотожності використовуються для встановлення границь послідовностей та знаходження похідних і інтегралів [8].

Задача 1.2.1. Довести тотожність

$$\cos\alpha\cos 2\alpha\cos 4\alpha\cdots\cos 2^n\alpha=\frac{\sin 2^{n+1}\alpha}{2^{n+1}\sin\alpha}.$$

Доведення. Умова 1) справджується для $n=0$, оскільки $\cos\alpha=\dfrac{\sin 2\alpha}{2\sin\alpha}$. (Умова 2) Припустимо, що тотожність справедлива для $n=k$, тобто

$$\cos\alpha\cos 2\alpha\cos 4\alpha\cdots\cos 2^k\alpha=\frac{\sin 2^{k+1}\alpha}{2^{k+1}\sin\alpha}.$$

Тоді для $n=k+1$ маємо

$$\cos\alpha\cos 2\alpha\cos 4\alpha\cdots\cos 2^k\alpha\cos 2^{k+1}\alpha=\frac{\sin 2^{k+1}\alpha\cos 2^{k+1}\alpha}{2^{k+1}\sin\alpha}=\frac{\sin 2^{k+2}\alpha}{2^{k+2}\sin\alpha}.$$

Задача 1.2.2. Довести, що

$$\sin x+\sin 2x+\ldots+\sin nx=\frac{\sin\dfrac{n+1}{2}x}{\sin\dfrac{x}{2}}\sin\frac{nx}{2}.$$

Доведення. Умова 1) справджується для $n = 1$, оскільки

$$\frac{\sin x}{\sin\dfrac{x}{2}}\sin\frac{x}{2} = \frac{2\sin\dfrac{x}{2}\cos\dfrac{x}{2}}{\sin\dfrac{x}{2}}\sin\frac{x}{2} = \sin x.$$

(Умова 2) Припустимо, що рівність справджується для $n = k$, тобто

$$\sin x + \sin 2x + \ldots + \sin kx = \frac{\sin\dfrac{k+1}{2}x}{\sin\dfrac{x}{2}}\sin\frac{kx}{2}.$$

Тоді для $n = k + 1$ маємо

$$\sin x + \sin 2x + \ldots + \sin kx + \sin(k+1)x = \frac{\sin\frac{k+1}{2}x}{\sin\frac{x}{2}}\sin\frac{kx}{2} + \sin(k+1)x =$$

$$= \frac{\sin\frac{k+1}{2}x}{\sin\frac{x}{2}}\sin\frac{kx}{2} + 2\sin\frac{k+1}{2}x\cos\frac{k+1}{2}x =$$

$$= \sin\frac{k+1}{2}x\left(\frac{\sin\frac{kx}{2} + 2\cos\frac{k+1}{2}x\sin\frac{x}{2}}{\sin\frac{x}{2}}\right) = \frac{\sin\frac{k+2}{2}x}{sin\frac{x}{2}}\sin\frac{k+1}{2}x$$

на підставі того, що

$$2\cos\frac{k+1}{2}x\sin\frac{x}{2} = \sin\frac{k+2}{2}x - \sin\frac{kx}{2}.$$

Задача 1.2.3. Довести, що $\frac{1}{2} + \cos x + \cos 2x + \ldots + \cos nx = \dfrac{\sin\frac{2n+1}{2}x}{2\sin\frac{x}{2}}$.

Доведення. Умова 1) справджується, оскільки для $n = 1$

$$\frac{1}{2} + \cos x = \frac{\sin\frac{x}{2} + (\sin\frac{3x}{2} - \sin\frac{x}{2})}{2\sin\frac{x}{2}} = \frac{\sin\frac{3x}{2}}{2\sin\frac{x}{2}}.$$

(Умова 2) Припустимо, що для $n = k$ справджується рівність

$$\frac{1}{2} + \cos x + \cos 2x + \ldots + \cos kx = \frac{\sin\frac{2k+1}{2}x}{2\sin\frac{x}{2}}.$$

Тоді

$$\frac{1}{2} + \cos x + \cos 2x + \ldots + \cos kx + \cos(k+1)x = \frac{\sin\frac{2k+1}{2}x}{2\sin\frac{x}{2}} + \cos(k+1)x =$$

$$= \frac{\sin\frac{2k+1}{2}x + 2\sin\frac{x}{2}\cos(k+1)x}{2\sin\frac{x}{2}} = \frac{\sin\frac{2k+1}{2}x + (\sin\frac{2k+3}{2}x - \sin\frac{2k+1}{2}x)}{2\sin\frac{x}{2}} =$$

$$= \frac{\sin\frac{2k+3}{2}x}{2\sin\frac{x}{2}}.$$

Задача 1.2.4. Знайти суму

$$S_n = \operatorname{arctg}\frac{1}{2} + \operatorname{arctg}\frac{1}{8} + \operatorname{arctg}\frac{1}{18} + \ldots + \operatorname{arctg}\frac{1}{2n^2}.$$

Розв'язання. Знайдемо частинні суми:

$S_1 = \operatorname{arctg}\frac{1}{2}$, $\quad S_2 = \operatorname{arctg}\frac{1}{2} + \operatorname{arctg}\frac{1}{8} = \operatorname{arctg}\frac{\frac{1}{2}+\frac{1}{8}}{1-\frac{1}{2}\cdot\frac{1}{8}} = \operatorname{arctg}\frac{2}{3}$,

$S_3 = \operatorname{arctg}\frac{2}{3} + \operatorname{arctg}\frac{1}{18} = \operatorname{arctg}\frac{\frac{2}{3}+\frac{1}{18}}{1-\frac{2}{3}\cdot\frac{1}{18}} = \operatorname{arctg}\frac{3}{4}$.

З огляду на отримані частинні суми можемо припустити, що

$$S_n = \operatorname{arctg}\frac{n}{n+1}.$$

Перевіримо справедливість припущення методом математичної індукції.
Умова 1 справджується для $n = 1, 2, 3$.
(Умова 2) Припустимо, що $S_n = \operatorname{arctg}\frac{n}{n+1}$. Тоді

$$S_{n+1} = \operatorname{arctg}\frac{n}{n+1} + \operatorname{arctg}\frac{1}{2(n+1)^2} = \operatorname{arctg}\frac{\frac{n}{n+1} + \frac{1}{2(n+1)^2}}{1 - \frac{n}{n+1}\cdot\frac{1}{2(n+1)^2}} = \operatorname{arctg}\frac{n+1}{n+2}.$$

Припущення справедливе і шукана сума має вигляд $S_n = \operatorname{arctg}\frac{n}{n+1}$.

ЗАДАЧІ ДЛЯ САМОСТІЙНОЇ РОБОТИ

1. Довести справедливість рівностей [5]:

а) $\sin x + 2\sin 2x + \ldots + n\sin nx = \dfrac{(n+1)\sin nx - n\sin(n+1)x}{4\sin^2\frac{x}{2}}$;

б) $\cos x + 2\cos 2x + \ldots + n\cos nx = \dfrac{(n+1)\cos nx - n\cos(n+1)x - 1}{4\sin^2\frac{x}{2}}$.

в) $\operatorname{arcctg} 3 + \operatorname{arcctg} 5 + \ldots + \operatorname{arcctg}(2n+1) =$
$= \operatorname{arctg} 2 + \operatorname{arctg}\frac{3}{2} + \ldots + \operatorname{arctg}\frac{n+1}{n} - n\operatorname{arctg} 1.$

2. Довести, що $A_n = \cos n\theta$, якщо відомо, що $A_1 = \cos\theta$, $A_2 = \cos 2\theta$ і для довільного $k \in \mathcal{N}^+$, більшого 2, справедливе співвідношення

$$A_k = 2\cos\theta A_{k-1} - A_{k-2}.$$

3. Довести, що $\frac{1}{2}\operatorname{tg}\frac{x}{2} + \frac{1}{2^2}\operatorname{tg}\frac{x}{2^2} + \cdots + \frac{1}{2^n}\operatorname{tg}\frac{x}{2^n} = \frac{1}{2^n}\operatorname{ctg}\frac{x}{2^n} - n\operatorname{ctg} x$, де $x \neq m\pi$.

4. Довести нерівність $|\sin\sum\limits_{k=1}^{n} x_k| \leq \sum\limits_{k=1}^{n}\sin x_k$, де $0 \leq x_k \leq \pi$.

1.3. Індукція у алгебрі

Метод математичної індукції в алгебрі зустрічається в доведеннях алгебричних тверджень майже з такою частотою, як і в теорії чисел та комбінаториці. Справа в тому, що велика кількість алгебричних конструкцій є дискретними, де цей метод з успіхом працює.

Задача 1.3.1. Довести, що $(\cos x+i\sin x)^n=\cos nx+i\sin nx$ для довільного $n\in\mathcal{N}^+$ (*формула Муавра*), де $i=\sqrt{-1}$ – уявна одиниця комплексних чисел. Користуючись цією формулою і тим, що $1+i=\sqrt{2}(\cos\frac{\pi}{4}+i\sin\frac{\pi}{4})$, знайти $(1+i)^{100}$.

Доведення. Очевидно, що умова 1) виконується для $n=1$.

(Умова 2) Припустимо, що для $n=k$ виконується рівність

$$(\cos x+i\sin x)^k=\cos kx+i\sin kx.$$

Тоді

$$\begin{gathered}(\cos x+i\sin x)^{k+1}=(\cos x+i\sin x)^k(\cos x+i\sin x)=\\ =(\cos kx+i\sin kx)(\cos x+i\sin x)=\\ =\cos x\cos kx+i\sin kx\cos x+i\cos kx\sin x-\sin kx\sin x=\\ =\cos kx\cos x+\sin\ \ kx\sin x+i(\sin x\cos kx-\sin kx\cos x)=\\ =\cos(k+1)x+i\sin(k+1)x.\end{gathered}$$

$$(1+i)^{100}=(\sqrt{2}(\cos\tfrac{\pi}{4}+i\sin\tfrac{\pi}{4}))^{100}=2^{50}(\cos 25\pi+i\sin 25\pi)=-2^{50}.$$

Задача 1.3.2. Довести, що

$$(\sqrt{3}-i)^n=2^n(\cos\frac{n\pi}{6}-\sin\frac{n\pi}{6}).$$

Доведення. Умова 1 справджується, оскільки $(\sqrt{3}-i)=2(\cos\frac{\pi}{6}-i\sin\frac{\pi}{6})$.

(Умова 2) Припустимо, що для $n=k$ справедлива рівність

$$(\sqrt{3}-i)^k=2^k(\cos\frac{k\pi}{6}-i\sin\frac{k\pi}{6}).$$

Тоді

$$(\sqrt{3}-i)^{k+1}=2^k(\cos\frac{k\pi}{6}-i\sin\frac{k\pi}{6})2(\cos\frac{\pi}{6}-i\sin\frac{\pi}{6})=$$

$$=2^{k+1}\left[(\cos\frac{k\pi}{6}\cos\frac{\pi}{6}-\sin\frac{k\pi}{6}\sin\frac{\pi}{6})-i(\sin\frac{k\pi}{6}\cos\frac{\pi}{6}+\cos\frac{k\pi}{6}\sin\frac{\pi}{6})\right]=$$

$$=2^{k+1}(\cos\frac{(k+1)\pi}{6}-i\sin\frac{(k+1)\pi}{6}).$$

Задача 1.3.3. Довести, що коли комплексне число $u = a+ib$ отримано в результаті виконання скінченної кількості операцій додавання, віднімання, множення і ділення над комплексними числами $x_1 = a_1 + ib_1, x_2 = a_2 + ib_2, \ldots, x_n = a_n + ib_n$, і коли ці самі операції виконати над спряженими числами $\bar{x}_1 = a_1 - ib_1, \bar{x}_2 = a_2 - ib_2, \ldots, \bar{x}_n = a_n - ib_n$, то дістанемо число $\bar{u} = a - ib$, спряжене до числа u.

Доведення. Перевірка умови 1) виконуються за допомогою перевірки справедливості твердження для кожної операції.

Нехай $x_1 = a_1 + b_1 i, \quad x_2 = a_2 + b_2 i$, Тоді

$x_1 + x_2 = (a_1 + a_2) + (b_1 + b_2)i = u,$

$\bar{x}_1 + \bar{x}_2 = (a_1 - b_1 i) + (a_2 - b_2 i) = (a_1 + a_2) - (b_1 + b_2)i = \bar{u}.$

$x_1 x_2 = (a_1 + b_1 i)(a_2 + b_2 i) = (a_1 a_2 - b_1 b_2) + (a_1 b_2 + b_1 a_2)i = u,$

$\bar{x}_1 \bar{x}_2 = (a_1 - b_1 i)(a_2 - b_2 i) = (a_1 a_2 - b_1 b_2) - (a_1 b_2 + b_1 a_2)i = \bar{u}.$

Для віднімання і ділення перевірка справедливості умови 1) виконується аналогічно.

(Умова 2) Нехай дано деякий вираз, побудований за допомогою вказаних чотирьох операцій над комплексними числами $x_1, x_2, \ldots, x_n$. Обчислення такого виразу зводиться до виконання вказаних дій над двома комплексними числами. Ці дії можна занумерувати. Наприклад, нехай

$$u = \frac{x_1 x_2 - x_3 x_4}{x_1 - x_2 + x_3}.$$

Обчислення u виконується за такою послідовністю дій:

1) $x_1 x_2 = u_1,$ 2) $x_3 x_4 = u_2,$ 3) $x_1 - x_2 = u_3,$

4) $u_3 + x_3 = u_4,$ 5) $u_1 - u_2 = u_5,$ 6) $u_5 : u_4 = u.$

Припустимо, що твердження справедливе для всіх виразів, побудова яких потребує не більше n операцій. Покажемо, що твердження справедливе і для виразів, які потребують $(n+1)$-ї операції.

Дійсно, $(n+1)$-а операція виконується над числами u_i і u_j, які обчислюються за допомогою виконання n операцій. На підставі припущення індукції при заміні $x_1, x_2, \ldots, x_n$ спряженими до них, числам u_i і u_j відповідають спряжені до них числа $\bar{u}_i$ і $\bar{u}_j$. Але тоді в результаті виконання $(n+1)$-ої операції над u_i і u_j отримуємо число u, якому відповідає спряжене число $\bar{u}$.

Задача 1.3.4. Довести, що для всіх натуральних $n \geq 3$ можливо виразити 1 як суму n різних дробів із чисельником 1.

Доведення. Умова 1) для $n = 3$ справджуються, оскільки

$$1 = \frac{1}{2} + \frac{1}{3} + \frac{1}{6}.$$

(Умова 2) Припустимо, що для $n = k > 3$ рівність справедлива, тобто

$$1 = \frac{1}{a} + \frac{1}{b} + \frac{1}{c} + \ldots + \frac{1}{k}$$

і дроби стоять в порядку спадання, тобто $\frac{1}{k}$ – найменший дріб.

Дріб $\frac{1}{k} = \frac{1}{k+1} + \frac{1}{k(k+1)}$, а звідси випливає, що оскільки обидва дроби $\frac{1}{k+1}$ і $\frac{1}{k(k+1)}$ менші $\frac{1}{k}$, то найменший дріб завжди можна розкласти на два різних дроби, кожен з яких менший за той, який розкладали. Тоді рівність справедлива для всіх натуральних значень $n \geq 3$.

Задача 1.3.5. Довести нерівність $(1+\alpha)^n \geq 1+n\alpha$, де $\alpha > -1$, $n \in \mathcal{N}^+$.

Доведення. Умова 1) для $n = 1$ справджується, оскільки, $1 + \alpha \geq 1 + \alpha$.

(Умова 2) Припустимо, що для $n = k$ нерівність справджується, тобто

$$(1 + \alpha)^k \geq 1 + k\alpha.$$

Тоді $(1 + \alpha)^{k+1} \geq (1 + k\alpha)(1 + \alpha)$ на підставі того, що $\alpha > -1$. Але

$$(1 + \alpha)^{k+1} \geq (1 + k\alpha)(1 + \alpha) = 1 + k\alpha + \alpha + k\alpha^2 \geq 1 + (k + 1)\alpha,$$

оскільки $k\alpha^2 > 0$. Отже, $(1 + \alpha)^n \geq 1 + n\alpha$ для всіх $n \in \mathcal{N}^+$.

Задача 1.3.6. Довести, що для довільного $n \in \mathcal{N}$ число $3^{2n+3}+40n-27$ кратне 64.

Доведення. Умова 1) для $n = 0$ справджується, оскільки $27 - 27 = 0$ кратне 64.

(Умова 2) Припустимо, що для $n = k > 0$ твердження справедливе, тобто $3^{2k+3} + 40k - 27$ ділиться на 64.

Доведемо справедливість твердження для $n = k + 1$.

$3^{2k+5} + 40(k + 1) - 27 = 9 \cdot 3^{2k+3} + 40k - 27 + 40 = (3^{2k+3} + 40k - 27)+$
$+(8 \cdot 3^{2k+3} + 40)$.

Перший доданок кратний 64 за припущенням. Доведемо, що й другий доданок кратний 64, або що те саме, що $3^{2k+3} + 5$ кратне 8.

Для доведення застосуємо метод математичної індукції.

Умова 1) для $m = 0$ виконується, оскільки 27 + 5=32 кратне 8.

(Умова 2) Припустимо, що твердження справедливе для $m = t$, тобто: $3^{2t+3} + 5$ кратне 8. Покажемо справедливість твердження і для $m = t + 1$. Дійсно, $3^{2t+5} + 5 = 9 \cdot 3^{2t+3} + 5 = (3^{2t+3} + 5) + 8 \cdot 3^{2t+3}$.

Перший доданок ділиться на 8 за припущенням індукції, а другий включає множник 8. Отже, $3^{2m+3} + 5$ кратне 8, а тому і перше твердження справедливе для всіх натуральних значень n.

В цьому доведенні використовувалася індукція в індукції. Такий спосіб доведення інколи називають подвійною індукцією. Цей спосіб поширюється і на потрійну індукцію, але якщо якесь твердження, було доведено подвійною математичною індукцією, то його потрібно довести ще раз за допомогою математичної індукції.

Задача 1.3.7. Довести, що для довільного $n \in \mathcal{N}^+$ справедливе твердження $2^{2n-1} - 9n^2 + 21n - 14$ кратне 27.

Доведення. Умова 1) для $n = 1$ справджується, оскільки $2-9+21-14 = = 0$ кратне 27.

(Умова 2) Припустимо, що для $n = k$ твердження справджується, тобто $2^{2k-1} - 9k^2 + 21k - 14$ кратне 27. Тоді

$$2^{2k+1} - 9(k+1)^2 + 21(k+1) - 14 = 4 \cdot 2^{2k-1} - 9k^2 - 18k - 9 + 21n + 21 - 14 = \\ = (2^{2k-1} - 9k^2 + 21k - 14) + (3 \cdot 2^{2k-1} - 18k + 12).$$

Перший доданок кратний 27 за припущенням індукції. Доведемо, що і другий доданок кратний 27. Виконаємо перетворення виразу $3 \cdot 2^{2k-1} - 18k + 12 = 3(2^{2k-1} - 6k + 4)$. Після скорочення доведемо, що $2^{2k-1} - 6k + 4$ кратне 9.

Умова 1) для $k = 1$ справджується, оскільки $2 - 6 + 4 = 0$ кратне 9.

(Умова 2) Припустимо, що для $k = t$ твердження справедливе, тобто $2^{2t-1} - 6t + 4$ кратне 9. Тоді

$2^{2t+1} - 6t - 2 = 4 \cdot 2^{2t-1} - 6t + 4 - 6 = (2^{2t-1} - 6t + 4) + (3 \cdot 2^{2t-1} - 6)$.

Перший доданок ділиться націло на 9 за припущенням індукції. Доведемо, що другий доданок теж кратний 9. Після скорочення на спільний дільник 3, потрібно показати що $2^{2t-1} - 2$ кратне 3.

Умова 1) для $t = 1$ справджується, оскільки $2 - 2 = 0$ кратне 3.

Припустимо, що твердження справедливе для $t = m$, $m > 1$, тобто $2^{2m-1} - 2$ кратне 3. Тоді $2^{2m+1} - 2 = 4 \cdot 2^{2m-1} - 2 = (2^{2m-1} - 2) + (3 \cdot 2^{2m-1})$.

Перший доданок ділиться націло на 3 за припущенням індукції, а другий складається з множників, один з яких – 3. Отже, обидва попередні твердження справедливі для всіх натуральних значень t та m відповідно, з чого випливає справедливість початкового твердження для всіх $n \in \mathcal{N}^+$.

Задача 1.3.8. Довести нерівність

$$\frac{x_1 + x_2 + \ldots + x_n}{n} \geq \sqrt[n]{x_1 x_2 \ldots x_n}.$$

Доведення. Умова 1) для $n = 2$ справджується, оскільки

$$\frac{x_1 + x_2}{2} - \sqrt{x_1 x_2} = \frac{(\sqrt{x_1} - \sqrt{x_2})^2}{2} \geq 0.$$

(Умова 2) Припустимо, що для $n > 2$ справджується нерівність

$$\frac{x_1 + x_2 + \ldots + x_n}{n} \geq \sqrt[n]{x_1 x_2 \ldots x_n}.$$

Тоді

$$\frac{x_1 + x_2 + \cdots x_n + x_{n+1}}{n+1} = \frac{n \cdot \frac{x_1 + x_2 + \cdots x_n}{n} + x_{n+1}}{n+1} \geq \frac{n \sqrt[n]{x_1 x_2 \cdots x_n} + x_{n+1}}{n+1}.$$

Покладемо $x_1x_2\cdots x_n = a^{n(n+1)}, x_{n+1} = b^{n+1}$ і, використовуючи попередню нерівність, дістанемо

$$\frac{x_1 + x_2 + \cdots x_n + x_{n+1}}{n+1} - \sqrt[n+1]{x_1x_2\cdots x_nx_{n+1}} \geq$$

$$\geq \frac{n\sqrt[n]{x_1x_2\cdots x_n} + x_{n+1}}{n+1} - \sqrt[n+1]{x_1x_2\cdots x_nx_{n+1}} =$$

$$= \frac{na^{n+1} + b^{n+1}}{n+1} - a^nb = \frac{na^{n+1} + b^{n+1} - na^nb - a^nb}{n+1} =$$

$$= \frac{1}{n+1}\left[na^n(a-b) - b(a^n - b^n)\right] = \frac{a-b}{n+1}[na^n - ba^{n-1} - b^2a^{n-2} - \ldots - b^n] =$$

$$= \frac{a-b}{n+1}[a^n - ba^{n-1} + a^n - b^2a^{n-2} + \ldots + a^n - b^n] =$$

$$= \frac{(a-b)^2}{n+1}[a^{n-1} + a^{n-2}(a+b) + a^{n-3}(a^2 + ab + b^2) +$$

$$+ \cdots + (a^{n-1} + a^{n-2}b + \cdots + b^{n-1})] \geq 0.$$

Нерівність доведено. Рівність досягається тоді, коли $x_1 = x_2 = \cdots = x_n$.

Задача 1.3.9. Нехай дано $x + y = m,\ xy = a,\ A_2 = m - \frac{a}{m-1}$,

$$A_3 = m - \cfrac{a}{m - \cfrac{a}{m-1}}, \quad A_4 = m - \cfrac{a}{m - \cfrac{a}{m - \cfrac{a}{m-1}}},$$

і т. д., тобто для $k > 1$

$$A_{k+1} = m - \frac{a}{A_k},\ m \neq 1, x \neq y.$$

Довести, що

$$A_n = \frac{(x^{n+1} - y^{n+1}) - (x^n - y^n)}{(x^n - y^n) - (x^{n-1} - y^{n-1})}. \tag{1.2}$$

Доведення. Умова 1) виконується для $n = 2$. Дійсно, за умовою

$$A_2 = m - \frac{a}{m-1} = (x+y) - \frac{xy}{(x+y)-1} = \frac{x^2 + y^2 + xy - x - y}{x + y - 1}.$$

За формулою 1.2 отримуємо

$$A_2 = \frac{(x^3 - y^3) - (x^2 - y^2)}{(x^2 - y^2) - (x - y)}.$$

Після скорочення на $x-y$, дістаємо $A_2=\frac{x^2+y^2+xy-x-y}{x+y-1}$, що і було потрібно.

(Умова 2) Нехай формула (1.2) справедлива для $n=k$, тобто

$$A_k=\frac{(x^{k+1}-y^{k+1})-(x^k-y^k)}{(x^k-y^k)-(x^{k-1}-y^{k-1})}. \tag{1.3}$$

Покажемо, що формула справедлива і для $n=k+1$, тобто

$$A_{k+1}=\frac{(x^{k+2}-y^{k+2})-(x^{k+1}-y^{k+1})}{(x^{k+1}-y^{k+1})-(x^k-y^k)}.$$

Справді,

$$A_{k+1}=m-\frac{a}{A_k}=(x+y)-\frac{xy}{A_k}.$$

На підставі рівності (1.3) отримуємо

$$A_{k+1}=(x+y)-\frac{xy[(x^k-y^k)-(x^{k-1}-y^{k-1})]}{(x^{k+1}-y^{k+1})-(x^k-y^k)}=\frac{(x^{k+2}-y^{k+2})-(x^{k+1}-y^{k+1})}{(x^{k+1}-y^{k+1})-(x^k-y^k)}.$$

Рівність доведено.

ЗАДАЧІ ДЛЯ САМОСТІЙНОЇ РОБОТИ

1, Довести, що для довільного натурального числа $n>1$

$$\frac{1}{\sqrt{1}}+\frac{1}{\sqrt{2}}+\ldots+\frac{1}{\sqrt{n}}>\sqrt{n}.$$

2. Довести , що

$$2^{n-1}(a^n+b^n)>(a+b)^n,$$

$a+b>0, a\neq b$ і $n\in\mathcal{N}^+$ більше 1.

3. Довести, що

$$(a_1+a_2+\ldots+a_n)^2=a_1^2+a_2^2+\ldots+a_n^2+2(a_1a_2+a_1a_3+\ldots+a_{n-1}a_n).$$

4. Довести нерівність Бернуллі

$$(1+x_1)(1+x_2)\cdots(1+x_n)\geq 1+x_1+x_2+\ldots+x_n,$$

де всі числа $x_1,x_2,\ldots,x_n$ дійсні, одного знаку і більші -1.

5. Довести нерівності

а) $n!\leq(\frac{n+1}{2})^n$, де $n>1$; б) $\frac{1}{2}\cdot\frac{3}{4}\cdot\ldots\cdot\frac{2n-1}{2n}\leq\frac{1}{\sqrt{2n+1}}$, де $n>1$;

в) $n^{n+1}>(n+1)^n$, де $n\geq 3$; г) $(2n)!<2^n(n!)^2$, де $n>1$.

5. Довести, що коли $x_1,x_2,\ldots,x_n$ невід'ємні дійсні числа, то

$$\sqrt[n]{x_1x_2\ldots x_n}\geq\frac{n}{\dfrac{1}{x_1}+\dfrac{1}{x_2}+\cdots+\dfrac{1}{x_n}}.$$

1.4. Індукція у геометрії і теорії графів

Застосування математичної індукції в задачах геометрії і теорії графів досить ефективне, але ускладнюється тим, що часто необхідно уявляти будову геометричних та графічних об'єктів.

Задача 1.4.1. Довести, що n різних прямих, які проходять на площині через одну точку, ділять площину на $2n$ частин.

Доведення. Умова 1) справджується для $n = 1$, оскільки одна пряма ділить площину на дві частини.

(Умова 2) Припустимо, що n прямих, які проходять через одну точку, ділять площину на $2n$ частин. Тоді $(n+1)$-ша пряма l поділить дві частини S_1 і S_2 площини на чотири частини (див. рис. 1.4.1). Але тоді кількість частин площини, на які вона розбивається $(n + 1)$-ою прямою l дорівнює $2n - 2 + 4 = 2(n + 1)$.

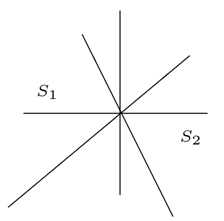
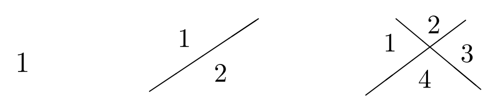


Рис. 1.4.1. До задачі 1.4.1.

Задача 1.4.2. Знайти залежність, за якою можна визначити на скільки частин ділять площину n попарно непаралельних прямих, які не проходять через одну точку [11].

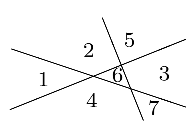
Розв'язання. Почнемо шукати залежність, починаючи з малих значень n. На рисунку 1.4.2 показано випадки для $n = 0, 1, 2$.



$L_0 = 1$ $\quad L_1 = 2$ $\quad L_2 = 4$

Рис. 1.4.2. Випадок $n = 0, 1, 2$

Для $n = 3$ за умовами задачі маємо



$L_3 = 4 + 3 = 7$

Рис.1.4.3. Випадок $n = 3$

Отже, $L_3 = 4 + 3 = 7$ як видно з рис.1.4.3. Якщо провести нову n-ту пряму, то кількість частин площини збільшиться на n тоді і тільки тоді, коли вона розбиває $n-1$ старих областей. А це відбувається тоді і тільки тоді, коли нова пряма перетинає старі прямі в $n-1$ різних точках. Але нова пряма може перетнути $n-1$ старих прямих в не більше ніж $n-1$ точках. Звідси випливає, що

$$L_n = L_{n-1} + n, \text{ \textit{де} } n > 0.$$

Враховуючи що $L_0 = 1$, отримуємо таку залежність

$L_0 = 1$,
$L_n = L_{n-1} + n$, якщо $n > 0$.

Розгортаючи цю залежність, дістаємо

$$\begin{aligned} L_n &= L_{n-1} + n = L_{n-2} + (n-1) + n = \\ &= L_{n-3} + (n-2) + (n-1) + n = \cdots = \\ &= L_0 + 1 + 2 + \ldots + (n-2) + (n-1) + n = 1 + \frac{n(n+1)}{2}. \end{aligned}$$

Доведемо методом математичної індукції справедливість отриманої залежності.

Умова 1) виконується для $n = 0$, оскільки $L_0 = 1 = \dfrac{0(0+1)}{2} + 1 = 1$.

(Умова 2) Припустимо, що для $n > 0$ рівність $L_n = L_{n-1} + n$ справедлива. Тоді

$$L_{n+1} = L_n + (n+1) = (\frac{n(n+1)}{2} + 1) + (n+1) = \frac{(n+1)(n+2)}{2} + 1.$$

Співвідношення доведено.

Задача 1.4.3. Довести, що n площин, які проходять через одну точку так, що ніякі три з них не проходять через одну пряму, ділять простір на $A_n = n(n-1) + 2$ частин.

Доведення. Умова 1) справджується для $n = 1$, оскільки одна площина ділить простір на дві частини $A_1 = n(n-1) + 2 = 2$.

(Умова 2) Припустимо, що твердження справедливе для $n > 1$, тобто $A_n = n(n-1) + 2$.

Нехай P – $(n+1)$-ша площина. Тоді ця площина перетинається з першими n площинами по деяких прямих. На підставі задачі 1.4.1. площина P розбивається n прямими, які проходять через одну точку, на $2n$ частин. Кожна частина утворює плоский кут з вершиною у цій точці.

Перші n площин ділять простір на багатогранні кути. Деякі з цих багатогранних кутів діляться площиною P на дві частини.

Спільною гранею двох частин є частина площини, обмежена двома променями, по яких площина P перетинається з гранями даного багатогранного кута, а це один із $2n$ плоских кутів, на які розбита площина P.

Це означає, що кількість багатогранних кутів, які розбиваються на дві частини площиною P, не перевищує величини $2n$.

Кожна із $2n$ частин, на які розбивається площина P, в результаті перетину її з першими n площинами, є спільною гранню двох багатогранних кутів і таким чином вона ділить багатогранний кут, утворений першими n площинами, на дві частини.

А це означає, що кількість багатогранних кутів, які розбиваються на дві частини площиною P, не може бути меншою $2n$.

Отже, площина P розбиває на дві частини точно $2n$ частин простору, утворених першими n площинами. Тому коли n площин розбивають простір на $n(n-1)+2$ частини, $n+1$-ша площина розбиває простір на $[n(n-1)+$ $+2]+2n = n(n+1)+2$ частини.

Задача 1.4.4. Зв'язний ациклічний граф T називається деревом.

Довести, що *в дереві* T *з* n *вершинами і* m *ребрами справедлива рівність* $m = n - 1$.

Доведення. Умова 1) справджується для дерева з єдиною вершиною ($n = 1$), оскільки таке дерево взагалі не має ребер і тому $m = n - 1 = 1 - 1 = 0$.

(Умова 2) Припустимо, що довільне дерево з $k < n$ вершинами має рівно $k - 1$ ребро.

Вилучивши одне ребро із T, отримаємо незв'язний граф, у якого два компоненти зв'язності, які теж є деревами. Позначимо ці дерева T_1 і T_2 і нехай вони мають n_1 і n_2 вершин відповідно. Оскільки $n_1 + n_2 = n$, то $n_1 < n$ і $n_2 < n$.

На підставі припущення індукції дерево T_1 має $n_1 - 1$ ребро, а T_2 має $n_2 - 1$ ребро. Тоді дерево T (з урахуванням видаленого ребра) матиме $(n_1-$ $-1) + (n_2 - 1) + 1 = n - 1$ ребро.

Задача 1.4.5. Зв'язний граф G називається ойлеровим, якщо існує замкнутий ланцюг, який включає всі ребра і кожне ребро входить до нього лише один раз. Такий ланцюг називають ойлеровим ланцюгом.

Довести, що *скінченний зв'язний граф* G *є ойлеровим тоді і тільки тоді, коли кожна вершина* G *має парний степінь (степенем вершини називають кількість ребер, для яких ця вершина є їхнім кінцем).*

Доведення. Нехай G має ойлерів цикл P, тоді при довільному проходженні через кожну вершину циклу P степінь цієї вершини збільшується на 2. А оскільки кожне ребро зустрічається в P тільки один раз, то кожна вершина повинна мати парний степінь.

Доведення у зворотному напрямку проводиться індукцією за числом ребер в графі G. Нехай v_1 – довільна вершина графа G. Оскільки степені всіх вершин парні, то, потрапивши у чергову вершину, завжди знайдеться ребро, яке ще не проходилося. Тому маршрут можна продовжити додаючи до нього це ребро. Нехай побудова закінчилася у вершині v_1 і знайдено маршрут P_1. P_1 буде циклом і коли йому належать всі ребра, то ойлерів цикл знайдено. В протилежному випадку вилучимо з G всі ребра циклу P_1, дістанемо граф G_1 степені всіх вершин якого знову парні. На підставі зв'язності графа G графи P_1 і G_1 матимуть принаймні одну спільну вершину v_2. Починаючи з v_2 будуємо цикл P_2 в графі G_1 подібно до того, як будувався цикл P_1. Нехай P_1' і P_1'' частини циклі P_1 від v_1 до v_2 і від v_2 до v_1 відповідно. Дістаємо новий цикл $P_3 = P_1' \cup P_2 \cup P_1''$. Якщо побудований цикл не ойлерів, то повторюючи побудову, прийдемо зрештою до ойлерового циклу.

Задача 1.4.6. Точка x площини S, на якій розміщений граф G, називається *диз'юнктною з* G, якщо x не збігається з жодною вершиною графа G і не належить жодному ребру цього графа. Дві точки x і y площини S називають еквівалентними, якщо вони диз'юнктні з G і їх можна з'єднати такою кривою, всі точки якої диз'юнктні з G (рис. 1.4.4 ліворуч). Введене відношення між точками площини є відношенням еквівалентності і за цим відношенням множина всіх точок площини S розбивається на класи еквівалентності, які називають *гранями графа* G. Наприклад, граф G, зображений на рис. 1.4.4 (праворуч), має чотири грані f_1, f_2, f_3, f_4 і грань f_4 – нескінченна його грань.

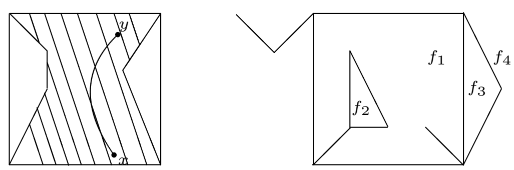


Рис. 1.4.4. Диз'юнктні точки графа та його грані

Довести, що *коли G – зв'язний плоский граф, в якому n, m і f відповідно число вершин, ребер і граней, то справедлива рівність $n + f = m + 2$.*

Доведення. (Умова 1) Якщо $m = 0$, то $n = 1$ (оскільки G зв'язний) і $f = 1$ (нескінченна грань), тобто в цьому випадку теорема має місце.

(Умова 2) Нехай теорема вірна для довільного графа G, який має $m-1$ ребро. Додамо до G нове ребро e (зауважимо, що це не є операція введення ребра в граф), тоді можливі такі випадки:

a) ребро e є петлею, і в цьому випадку число граней збільшується на одну, а число вершин залишається незмінним;

b) ребро e з'єднує дві різні вершини в G, і в цьому випадку одна з граней графа G розпадається на дві, число граней зростає на одиницю, а число вершин залишається незмінним;

c) ребро e інцидентне лише одній з вершин графа G, і в цьому випадку число його вершин зростає на одиницю, а число граней залишається незмінним.

У кожному з цих випадків твердження залишається справедливим, а оскільки наведеними випадками вичерпуються всі можливості, то цим воно доведено повністю.

Задача 1.4.7. Барицентром (центром мас) системи точок $a_1, a_2, \ldots, a_n$ називається точка G, для якої справедлива векторна рівність

$$\overrightarrow{Ga_1} + \overrightarrow{Ga_2} + \ldots + \overrightarrow{Ga_n} = \overrightarrow{0}. \tag{1.4}$$

Довести, що довільна скінченна система точок має єдиний барицентр [9].

Доведення. Покажемо існування барицентра методом математичної індукції.

(Умова 1) Для $n = 2$ барицентром системи a_1, a_2 є середина відрізка $\overline{a_1 a_2}$.

(Умова 2) Припустимо, що твердження правильне для $n = k$, тобто існує точка G' така, що

$$\overrightarrow{G'a_1} + \overrightarrow{G'a_2} + \ldots + \overrightarrow{G'a_k} = \overrightarrow{0}. \tag{1.5}$$

Розглянемо точку G, яка ділить напрямлений відрізок $\overrightarrow{a_{k+1}G'}$ у відношенні k, тобто точку, для якої справедлива векторна рівність $\overrightarrow{a_{k+1}G} = k\overrightarrow{GG'}$. Останню рівність можна переписати у вигляді

$$\overrightarrow{a_{k+1}G} = \overrightarrow{Ga_1} + \overrightarrow{a_1G'} + \overrightarrow{Ga_2} + \overrightarrow{a_2G'} + \ldots + \overrightarrow{Ga_k} + \overrightarrow{a_kG'}$$

або

$$\overrightarrow{G'a_1} + \overrightarrow{G'a_2} + \ldots + \overrightarrow{G'a_k} = \overrightarrow{Ga_1} + \overrightarrow{Ga_2} + \ldots + \overrightarrow{Ga_k} + \overrightarrow{Ga_{k+1}}.$$

Але з (1.5) випливає, що ліва частина останньої рівності дорівнює $\overrightarrow{0}$. Тому $\overrightarrow{Ga_1} + \overrightarrow{Ga_2} + \ldots + \overrightarrow{Ga_{k+1}} = \overrightarrow{0}$.

Отже, точка G є барицентром системи точок $a_1, a_2, \ldots, a_k, a_{k+1}$. Тоді, на підставі припущення математичної індукції, дане твердження справджується для довільної скінченної кількості точок.

Тепер доведемо єдиність. Нехай крім рівності (1.4) справедлива і рівність

$$\overrightarrow{Oa_1} + \overrightarrow{Oa_2} + \ldots + \overrightarrow{Oa_n} = \overrightarrow{0}. \tag{1.6}$$

Віднявши від рівності (1.4) рівність (1.6), одержимо

$$(\overrightarrow{Ga_1} - \overrightarrow{Oa_1}) + (\overrightarrow{Ga_2} - \overrightarrow{Oa_2}) + \ldots + (\overrightarrow{Ga_n} - \overrightarrow{Oa_n}) = \overrightarrow{0} = n\overrightarrow{GO}.$$

Звідки $G = O$. Єдиність а разом з ним і все твердження доведено.

ЗАДАЧІ ДЛЯ САМОСТІЙНОЇ РОБОТИ

1. Довести, що n попарно не паралельних прямих, які лежать на одній площині і жодні 3 з яких не проходять через одну точку, поділяють площину на $\frac{1}{2}(n^2+n+2)$ частин.

2. Точки $A_1, A_2, \ldots, A_n$ розташовані на площині так, що кожні 4 з них лежать у вершині опуклого чотирикутника. Довести, що всі точки лежать у вершинах опуклого n-кутника [4].

3. На площині задано набір з n векторів, довжина кожного з яких не перевищує 1. Довести, що замінивши деякі вектори цього набору на протилежні, можна одержати набір векторів, сума яких має довжину, що:

а) не перевищує $\sqrt{n}$; б) не перевищує $\sqrt{2}$ [9].

4. Довести, що кількість діагоналей d_n випуклого n-кутника можна обчислити за формулою $\dfrac{n(n-3)}{2}$.

5. Довести, що n кіл, які лежать в одній площині, поділяють площину не більш ніж на $n^2 - n + 2$ частин.

6. Знайти суму внутрішніх кутів n-кутника (не обов'язково випуклого).

1.5. Індукція у програмуванні

Використання методу математичної індукції у програмуванні найчастіше зустрічається в задачах доведення правильності алгоритмів і програм. Але варто зазначити користь від доведених в попередніх підрозділах тотожностей і рівностей, яка чітко проглядається при програмуванні чисельних методів. Розглянемо приклад для ілюстрації сказаного.

Нехай потрібно написати програму обчислення частинної суми ряду

$$1 \cdot 2 + 2 \cdot 3 + 3 \cdot 4 + \ldots + (n-1)n$$

задачі 1.1.3 з першого підрозділу. Програміст, не знайомий з методами обчислення таких сум, або їм подібних, напише програму приблизно такого вигляду:

```
proc : sumofprods(n)
  sum, i, j, n : integer;
  begin
    sum := 0;
    for i := 2 to n do
      j := (i − 1) ∗ i;
      sum := sum + j;
    od
    output(sum);
  end
end sumofprods
```

Але цю саму суму обчислить програма з одним оператором

$proc : sumofprods(n)$
$sum, s, n : integer;$
begin
$s := \frac{(n-1)n(n+1)}{3}.$
end
$end\ sumofprods$

Ефективність другої програми очевидна: перша програма під час обчислень виконає $n-1$ операцій множення і $n-1$ операцій додавання, в той час як другій програмі для обчислення того самого результату знадобиться лише одна операція ділення (яка за складністю така сама як операція множення), дві операції множення і дві операції додавання. Така сама ситуація з обчисленням суми ряду у задачах 1.1.4, 1.1.5, 1.1.6 з підрозділу 1.1.

Задача 1.5.1. Довести правильність нижченаведеного алгоритму, який знаходить максимальний елемент у скінченній множині A натуральних чисел, які відмінні від нуля.

$\mathbf{MAX(A, n)}$
$i, m : integer;$
begin
$i := 0;\ m := 0;$
while $i < n$ do
$i := i + 1;$
$m := \max(m, A(i));$
od
$print(m);$
end

Доведення. Дія такого алгоритму на множині $A = \{a_1 = 4, a_2 = 7, a_3 = 3, a_4 = 8, a_5 = 9\}$ показана в таблиці

i	m	$i < 5$
0	0	1
1	4	1
2	7	1
3	7	1
4	8	1
5	9	0

Отже, $m = \max(A) = 9$.

Тепер покажемо, що цей алгоритм правильно працюватиме на скінченній множині, відмінних від нуля натуральних чисел, потужність якої може бути довільною.

Нехай множина A має n елементів $a_1, a_2, \ldots, a_n$ і m_k означає значення змінної m після k-го виконання тіла циклу.

(*База індукції* $n = 1$). Якщо $n = 1$, то маємо одноелементну множину $A = \{a_1\}$ довжини 1. Тоді тіло циклу виконається лише один раз і m присвоїться значення $\max(0, a_1) = a_1$. В цьому випадку результат роботи алгоритму буде правильний.

(*Крок індукції*). Якщо після k-ї ітерації цикла m_k є максимальним елементом підмножини $\{a_1, a_2, \ldots, a_k\}$, то після $k+1$-ї ітерації цикла m_{k+1} дорівнюватиме $\max(m_k, a_{k+1})$, а це буде максимальний елемент підмножини $\{a_1, a_2, \ldots, a_k, a_{k+1}\}$.

Отже, на підставі методу математичної індукції отримуємо, що алгоритм буде правильно працювати для скінченної множини відмінних від нуля натуральних чисел довільної потужності.

Задача 1.5.2. Довести правильність нижченаведеного алгоритму.

Квадрат(n)

Вхід: натуральне число $n \in \mathcal{N}^+$.

Вихід: квадрат n^2 числа n.

Метод:

$sq := 0;$
for $i := 1$ to n do
  $sq := sq + 2i + 1;$
od

Доведення. Нехай $P(n)$ означає умову $sq = n^2$ після n-го виконання тіла циклу, а sq_k – значення змінної sq після k-го виконання тіла циклу.

Умова 1) набуває вигляду $sq_1 = 1^2$ і очевидно справджується.

(Умова 2) Нехай $sq_k = k^2$, тоді після $(k+1)$-го виконання тіла циклу маємо

$$sq_{k+1} = sq_k + 2(k+1) + 1 = k^2 + 2k + 1 = (k+1)^2.$$

На підставі того, що умова $P(n)$ справджується правильність алгоритму встановлена.

Зауваження. *В цьому алгоритмі цикл* **for** *обмежується певною кількістю ітерацій. У тому випадку, коли кількість ітерацій циклу завчасно невідома, як в циклі* **while ... do**, *то при доведенні правильності алгоритму методом індукції потрібно припускати, що кількість ітерацій обмежена. Після цього потрібно показати, що ця кількість дійсно обмежена.*

Задача 1.5.3. Довести правильність нижченаведеної програми P_1, яка обчислює результат піднесення числа x до степеня y (x^y), де $x, y \in \mathcal{N}$:

1. $i := 0; z := 1$
2. if $y = 0$ then stop
3. if $x = 0$ then ($z := 0$; stop)
4. repeat
5. $i := i + 1$
6. $z := z \cdot x$
7. until $i = y$

Необхідно довести тотальну правильність цієї програми, тобто довести, що програма завжди завершує свою роботу при передумовах, що $x, y \in \mathcal{N}$ і представляються в пам'яті комп'ютера, і що після її завершення стають істинні постумови.

Покажемо спочатку, що при виконанні передумов нескінченні цикли в програмі не виникають. Нескінченний цикли може виникнути тільки в області дії оператора *repeat*. Якщо умови оператора 2 або 3 справджуються, то виконання програми обмежується виконанням частини then відповідного оператора (оскільки після кожного із них програма зупиняється). Залишилося перевірити єдиний цикл в програмі P_1.

Завершення циклу (оператори $4 - 7$) залежить від виконання умови $i = y$. Із передумови випливає, що $y \geq 0$ і коли виконується умова $y = 0$, то виконання оператора 2 завершується оператором 8 без входу в цикл. В решті випадків попадання в цикл можливе. Кожне виконання операторів тіла циклу приводить до збільшення значення змінної i на одиницю і обов'язково рано чи пізно воно зрівняється за значенням з y, оскільки це число залишається незмінним при роботі програми. Отже, виконання циклу обов'язково завершується при довільному вхідному значенні y, яке задовольняє передумову і яке представляється в даному комп'ютері.

Доведемо тепер часткову правильність програми. Розглянемо можливі випадки:

а) випадок $y = 0$. Тоді виконання оператора 1 закінчиться присвоюванням $z := 1$, а виконання оператора 2 приводить до виконання оператора 8, тобто до завершення програми. Отже, при $y = 0$ маємо $z = 1$ і це є вірним результатом;

б) випадок $y > 0$. При $x = 0$ маємо $z = 0$ і програма закінчується. Але із $x = 0$ і $y > 0$ випливає $z_0 = x^y = 0^y = 0$.

При попаданні в цикл змінній i буде присвоюватися значення 1. Потім виконується присвоювання $z := 1 * x$, а звідси отримуємо, що $z_1 = x = x^1$ після першої ітерації циклу. Отже, $z_1 = x^1$ вірне на мінімальному значенні індекса циклу.

Припустимо, що після n-ї ітерації циклу вірно $z_n = x^n$. Покажемо, що $z_{n+1} = x^{n+1}$ після $n + 1$-го виконання тіла циклу. На $n + 1$-ї ітерації циклу після виконання 5 маємо $i = n + 1$, а в наступному операторі виконується присвоювання $z := z_n * x$. Оскільки це $n+1$-ше присвоювання, то маємо $z_{n+1} = z_n * x = x^n * x = x^{n+1}$ згідно з припущенням індукції, а це і є те, що потрібно було довести.

Отже, твердження $(\forall i \in \mathcal{N})(z_i = x^i)$ теж істинне.

Чи завершується цим доведення тотальної правильності? На жаль, ні. Залишається ще один момент, який пов'язаний з питанням про представлення цілих чисел в комп'ютерах.

Нагадаємо, що програма P_1 забезпечує правильну відповідь за умови, що вхідні дані x і y представляються в даному комп'ютері. Але в нашому випадку можливість представлення x і y не гарантує можливості представлення результату x^y. Чи можна після цього стверджувати, що програма P_1 правильна? Строго кажучи, цього стверджувати не можна, оскільки це суперечить означенню тотальної правильності програм.

Висновок, який випливає із сказаного, невтішний: не існує програми P_1, яка робить істинним твердження

$$\{\forall x \forall y (x, y \in \mathcal{N})\} P_1 \{z = x^y\}.$$

Необхідно змиритися з тим, що довести потрібну властивість програми P_1 не вдалося. Більше того, з'ясувалося що це взагалі неможливо зробити. В тому вигляді, в якому задача була поставлена, вона не має розв'язку. Але виконаний аналіз не був безплідним. По-перше, було встановлено, що при деяких (а не всіх) вхідних даних програма P_1 працює правильно. По-друге, була з'ясована причина, із-за якої не вдалося провести доведення правильності програми P_1. По-третє, стає зрозумілим, як потрібно доповнити передумови з тим, щоб забезпечити істинність постумови $z = x^y$. Для цього необхідно умову "x^y представляється" ввести в передумови, якщо є необхідність одержати тотальну правильність програми P_1:

$$\{\forall x \forall y (x, y \in \mathcal{N}) \ \wedge \ x^y \text{ представляється}\} P_1 \{z = x^y\}.$$

Тепер можна сказати, що істинність даного твердження доведена.

ЗАДАЧІ ДЛЯ САМОСТІЙНОЇ РОБОТИ

1. Довести правильність бінарного алгоритму обчислення НСД:

   Бінарний-НСД(m, n)
   Вхід: integer m, n.
   Вихід: НСД(m, n).
   Метод:
   1. $d := 1$;
   2. while m and n are even do
      2.1. $(m := m/2;\ n := n/2; d := 2d)$;

od
3. while $m \neq 0$ do
3.1. while m is even do $m := m/2$; od
3.2. while n is even do $n := n/2$; od
3.3. $t := |m - n|/2$;
3.4. if $m \geq n$ then $m := t$ else $n := t$;
od
4. return $(d \cdot n)$.
end Бінарний-НСД(m, n**)**

2. Довести правильність алгоритму Уоршалла

Найкоротші-шляхи(G, w**)**
Вхід: Орграф $G = (V, E)$ у вигляді матриці суміжностей M_G, елементи якої $m(i, j) = w(i, j)$, і якщо $(v_i, v_j) \notin E$, то $w(i, j) = \infty$, $i, j = 1, \ldots, n$.
Вихід: Матриця C, елементи якої $c(i, j)$ дорівнюють найкоротшим шляхам в орграфі між вершинами v_i і v_j, $i, j = 1, 2, \ldots, n$.
Метод:
for $i := 1$ to n do
for $j := 1$ to n do
$c(i, j) := m(i, j)$;
for $k := 1$ to n do
for $i := 1$ to n do
for $j := 1$ to n do
$c(i, j) := \min(c(i, j), c(i, k) + c(k, j))$;
end Найкоротші-шляхи

3. Довести правильність алгоритму обходу (в глибину) вершин графа

DFS-Connected-Component(G)
Вхід: Неорієнтований граф $G = (V, E)$.
Вихід: Множина помічених вершин графа $G = (V, E)$.
Метод:
$c := 0$;
for all vertices $v \in V$ do
$visited(v) := false$; $finished(v) := false$;
$p(v) := nil$
od;
for all vertices $v \in V$ do
if *not* $visited(v)$ then
$c := c + 1$; $DFS(v, c)$
fi;
od;
end DFS-Connected-Component

DFS(v, c)
$visited(v) := true$;
$component(v) := c$;

```
    for all vertices w ∈ adj(v) do
      if not visited(w) then
        p(w) := v;   DFS(w, c)
      fi;
    od;
    finished(v) := true;
end DFS
```

У цьому алгоритмі використовується масив $p(v)$ для запам'ятовування предка вершини v в DFS-дереві. В кожному компоненті зв'язності тільки корінь буде мати порожній список предків в DFS-дереві.

Підсумки. Якщо підвести підсумки використання методу математичної індукції, то з наведених прикладів чітко вимальовуються вимоги до множини, на якій діє цей метод. Першою вимогою є існування у такій множині найменшого елемента для обґрунтування базису індукції. Оскільки не в кожній множині ця умова виконується (наприклад, у множині цілих чисел умова не виконується), то необхідно з'ясувати у яких множинах, крім множини натуральних чисел може застосовуватися метод індукції. Друга вимога потребує дискретної упорядкованості множини для обґрунтування індуктивного переходу.

Обидві ці умови пов'язані з відношенням порядку, визначеним на множині, і застосування чи не застосування методу математичної індукції залежить від властивостей цього порядку.

У зв'язку з вказаними вимогами розглянемо відношення порядку та його варіації на множинах, яке дає можливість застосування методу математичної індукції.

Розділ 2

ОСНОВИ МЕТОДУ МАТЕМАТИЧНОЇ ІНДУКЦІЇ

Відношення часткового порядку – одне з фундаментальних відношень у математиці і є основою для введення інших відношень порядку. Зокрема, воно визначає типи упорядкованих множин, на яких застосовний метод математичної індукції.

2.1. Відношення часткового порядку

Означення 2.1. *Бінарне відношення O, 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:*

а) $aOa \qquad (i_A \subseteq O)$ (*рефлексивність*);

б) aOb *і* $bOc \Rightarrow aOc$ $(O^2 \subseteq O)$ (*транзитивність*);

в) aOb *і* $bOa \Rightarrow a = b$ $(O \cap O^{-1} \subseteq i_A)$ (*антисиметричність*).

Частковий порядок на множині A, як правило, позначають символом $\leq$, а саму частково упорядковану множину A називають скорочено **чум** і позначають $(A, \leq)$.

Означення 2.2. *Транзитивне і антирефлексивне відношення називається відношенням строгого порядку. Це відношення позначають $<$.*

Транзитивне і рефлексивне відношення називається відношенням квазіпорядку. Це відношення позначають $\preceq$.

З кожним відношенням часткового порядку $\leq$ пов'язане відношення строгого порядку $<$ (цей порядок називають строгою частиною відношення $\leq$):

$$a < b \Leftrightarrow a \leq b \text{ і } a \neq b.$$

Навпаки, з кожним відношенням строгого порядку $<$ пов'язане відношення часткового порядку $\leq$:

$$a \leq b \Leftrightarrow a < b \text{ або } a = b.$$

З кожним відношенням квазіпорядку $\preceq$ пов'язані відношення строгого порядку $<$ і еквівалентності $\sim$:

$$(a < b) \Leftrightarrow a \preceq b \text{ і } \neg(b \preceq a), \qquad (a \sim b) \Leftrightarrow a \preceq b \text{ і } b \preceq a.$$

Довільне відношення квазіпорядку $\preceq$, задане на множині A, індукує відношення часткового порядку $\leq$ на фактор-множині $A/\sim$:

$$[a]_\sim \leq [b]_\sim \Leftrightarrow a \preceq b.$$

Дійсно, оскільки відношення $\preceq$ транзитивне і рефлексивне, то і відношення $\leq$ теж буде транзитивним і рефлексивним. Покажемо антисиметричність. $[a]_\sim \leq [b]_\sim$ і $[b]_\sim \leq [a]_\sim$ означають, що $a \preceq b$ і $b \preceq a$. Але звідси випливає, що $a \sim b$, тобто, що $[a]_\sim = [b]_\sim$.

Властивості чум. Першою важливою властивістю чум є принцип двоїстості.

Теорема 2.1 (принцип двоїстості). *Відношення, обернене до відношення часткового порядку, теж буде відношенням часткового порядку.*

Доведення. Нехай $\leq^{-1}$ – відношення, обернене до відношення $\leq$.
а) Оскільки $i_A \subseteq \leq$, то $i_A = i_A^{-1} \subseteq \leq^{-1}$.
l) Якщо $\leq * \leq \subseteq \leq$, то $\leq^{-1} * \leq^{-1} = (\leq * \leq)^{-1} \subseteq \leq^{-1}$.
в) Якщо $\leq \cap \leq^{-1} \subseteq i_A$, то $\leq^{-1} \cap \leq \subseteq i_A$ і $\leq^{-1} \cap (\leq^{-1})^{-1} \subseteq i_A$. ■[1]

Означення 2.3. *Відношення часткового порядку* $\leq^{-1}$ *називають* **двоїстим до** *відношення часткового порядку* $\leq$.

Відношення $\leq^{-1}$ позначають $\geq$ і $a \leq^{-1} b$ означає $a \geq b$. Якщо $a \leq b$ або $b \leq a$, то a, b називають елементами, що **порівнюються відносно порядку** $\leq$.
З принципу двоїстості випливає, що коли в якому-небудь твердженні про чум замінити частковий порядок на двоїстий до нього порядок, то одержане твердження теж буде справедливим.

[1] Знак ■ означає закінчення доведення, а знак ♠ – закінчення прикладу.

Означення 2.4. *Елемент x із множини $(A, \leq)$ називається* **мінімальним (максимальним) елементом** A*, якщо для довільного елемента a із A, що порівнюється з x, має місце нерівність $x \leq a$ $(x \geq a)$.*

Елемент x із A називається **найбільшим (найменшим)***, якщо $(\forall a \in \in A)$ $x \geq a$ $(x \leq a)$.*

З цього означення очевидним чином випливає, що коли a і a' різні мінімальні (максимальні) елементи деякої чум $(A, \leq)$, то вони не порівнюються між собою відносно порядку $\leq$.

Теорема 2.2. *В довільній чум $(A, \leq)$ існує не більше одного найменшого (а на підставі принципу двоїстості і найбільшого) елемента.*

Доведення. Припустимо, що a і b – два найменші елементи в множині A, тоді $a \leq b$ на підставі того, що a найменший елемент і $b \leq a$ на підставі того, що b найменший елемент. Але тоді із антисиметричності відношення $\leq$ випливає, що $a = b$ ■

Якщо $(A, \leq)$ чум, то множина її елементів $[a, b] = \{s \in A \mid a \leq s \leq b\}$ називається **інтервалом**. Зрозуміло, що $a, b \in [a, b]$ і якщо a, b найменший і найбільший елементи множини A відповідно, то $A = [a, b]$.

Означення 2.5. *Підмножину Q чум $(A, \leq)$, всі елементи якої порівнюються між собою, називають ланцюгом чум. А підмножину P чум $(A, \leq)$ називається* **антиланцюгом** *чум, якщо довільні два її елементи x і y, де $x \neq y$, не порівнюються між собою.*

Висотою *чум $P = (A, \leq)$ називається максимальна потужність її ланцюга, а* **шириною** *— максимальна потужність її антиланцюга.*

Якщо $\leq$ і $\leq'$ часткові порядки, задані на множині A, то $\leq'$ називається *розширенням* $\leq$, якщо $\leq \subseteq \leq'$.

Множина максимальних (мінімальних) елементів чум $(A, \leq)$ позначається $MAX(A, \leq)$ $(MIN(A, \leq))$. Підмножини $MAX(A, \leq)$ і $MIN(A, \leq)$ завжди визначають антиланцюги, які ніколи не перевищують ширини $(A, \leq)$.

Нехай $(A, \leq)$ і $(B, \leq')$ – дві чум. Відображення $f : A \to B$ називається **вкладенням**, або **ізотонним відображенням** $(A, \leq)$ в $(B, \leq')$, якщо із $a \leq b$ в $(A, \leq)$ випливає $f(a) \leq' f(b)$ в $(B, \leq')$. Відображення $f : A \to B$ називають **ізоморфізмом чум**, якщо воно взаємно однозначне і $a \leq b \Leftrightarrow f(a) \leq f(b)$ для довільних елементів $a, b \in A$. Ізоморфні множини вважають рівними. Ізоморфізм чум $(A, \leq)$ $f : A \to A$ називають **автоморфізмом**.

З визначення ізоморфізму чум випливає, що мінімальні і максимальні елементи ізоморфних чум повинні відображатися один в одного при ізоморфізмі. Звідси випливає, що відображення $f : \mathcal{N} \to \mathcal{N}$ яке має вигляд

$f(n) = n + 1$, не буде автоморфізмом, оскільки $f(0) = 1$, а мало б бути $f(0) = 0$.

Існує поняття інверсно ізотонних й інверсно ізоморфних чум. Відображення $f : A \to B$ називають **інверсно ізотонним**, якщо із $a \leq b$ в A випливає $f(b) \leq' f(a)$ в B. Відображення $f : A \to B$ називають **інверсним ізоморфізмом**, якщо воно взаємно однозначне і $a \leq b \Leftrightarrow f(b) \leq' f(a)$ для довільних $a, b \in A$.

Справедливі такі твердження.

Теорема 2.3. *Відображення f чум $(A, \leq)$ на чум $(B, \leq')$ є ізоморфізмом, якщо для довільних $a, b \in A$ нерівність $a \leq b$ має місце тоді і тільки тоді, коли $f(a) \leq' f(b)$.*

Доведення. Для доведення досить показати, що із $f(a) = f(b)$ випливає $a = b$, де $a, b \in A$. Але за умовою із $f(a) = f(b)$ випливає $a \leq b$ і $b \leq a$ і на підставі антисиметричності відношення часткового порядку отримуємо $a = b$. ■

Теорема 2.4 (Ділуорс). *Якщо $(A, \leq)$ – чум ширини n, то існує розбиття множини $A = C_1 \cup C_2 \cup \ldots \cup C_n$ таке, в якому кожний клас C_i є ланцюгом.*

Доведення виконується методом математичної індукції за потужністю $|A|$ множини A.

База індукції для $|A| = 1$ виконується очевидним чином.

(*Крок індукції*) Припустимо, що твердження має місце для $|A| < k$. Розглянемо $P = (A, \leq)$, де $|A| = k$. Можна припустити, що ширина $P = (A, \leq)$ більша 1.

Виберемо $x \in MAX(P = (A, \leq))$ і $y \in MIN(P = (A, \leq))$ так, що $y \leq x$. Нехай $Q = (A \setminus \{x, y\}, \leq)$. Якщо ширина Q менша n, то за припущенням індукції існує розбиття Q на не більше ніж n ланцюгів, які разом з ланцюгом $\{y, x\}$ складають розбиття A на (щонайбільше) n ланцюгів.

Виберемо n-елементний антиланцюг $V = \{a_1, a_2, \ldots, a_n\}$ в Q. Тоді нехай $U = \{u \in A \mid u \geq a_i$ для деякого $a_i \in V\}$ і $D = \{d \in A \mid d \leq a_j$ для деякого $a_j \in V\}$. Отже, існує ланцюг розбиттів $U = C_1' \cup C_2' \cup \ldots \cup C_n'$ і $D = C_1'' \cup C_2'' \cup \ldots \cup C_n''$. Ці ланцюги можна вибрати так, що $a_i \in C_i' \cap C_i''$ для $i = 1, 2, \ldots, n$. А це означає, що $C_i = C_i' \cup C_i''$ є ланцюгом для кожного i, що дає шукане розбиття $A = C_1 \cup C_2 \cup \ldots \cup C_n$. ■

Теорема Ділуорса має тривіальну двоїсту версію.

Теорема 2.5 (двоїста теоремі Ділуорса). *Якщо $(A, \leq)$ чум висоти n, то існує розбиття $A = A_1 \cup A_2 \cup \ldots \cup A_n$, в якому кожне A_i є антиланцюгом.*

Доведення. Покладемо $A_1 = MAX((A, \leq))$. Тоді покладемо $A_{i+1} = MAX(P_i)$, де чум P_i отримана з множини A шляхом вилучення антиланцюгів $A_1, A_2, \ldots, A_i$. ■

Говоримо, що чум задовольняє **умову мінімальності**, якщо довільна її непорожна підмножина має мінімальний елемент.

Зазначимо, що чум може мати декілька і навіть нескінченно багато як максимальних, так і мінімальних елементів.

Якщо чум M задовольняє умову мінімальності і кількість мінімальних елементів довільної її підмножини скінченна, то M називають *добре частково упорядкованою* множиною. Зокрема, якщо кожна непорожна підмножина M має єдиний мінімальний елемент, то M називають *добре упорядкованою множиною.*

Теорема 2.6. *Нехай M – чум з умовою мінімальності і B підмножина M. Якщо з того, що B включає всі елементи із M, строго менші за a, випливає що B включає і сам елемент a, то $B = M$.*

Доведення. Припустимо, що $B \neq M$. Тоді $M \setminus B \neq \emptyset$ і має мінімальний елемент x. Оскільки x мінімальний, то кожний попередник елемента x не може належати множині $M \setminus B$. Але тоді всі попередники елемента x належать множині B і сам елемент x теж належить B. Суперечність із тим, що $x \in M \setminus B$ і $x \in B$. ■

Ця теорема носить назву **узагальненого принципу математичної індукції**.

Неважко збагнути, що чум з умовою мінімальності приводить до такої умови.

Умова обриву спадних ланцюгів: *довільний строго спадний ланцюг чум $a_1 > a_2 > \ldots > a_n > \ldots$ стабілізується на скінченному індексі, тобто знайдеться такий індекс n, що $a_n = a_{n+1} = \ldots$.*

Чум з умовою обриву спадних ланцюгів називають **фундованою множиною** (формальне означення фундованості буде наведено далі).

Умова індуктивності. *Усі елементи чум M задовольняють деяку властивість P, якщо цю властивість задовольняють усі мінімальні елементи із M (коли вони існують) і якщо із справедливості P для всіх елементів строго менших за елемент a випливає справедливість властивості P і для самого елемента a.*

Добре відомий такий результат.

Теорема 2.7. *Умови мінімальності, індуктивності і фундованості еквівалентні між собою [6].*

З теореми Ділуорса випливає, що чум ширини n тоді і тільки тоді задовольняє умову мінімальності, коли всі її ланцюги добре упорядковані множини. А з принципу двоїстості випливає, що в чум можна перейти до двоїстого відношення порядку. Якщо чум задовольняє умову мінімальності, то взявши двоїстий порядок, дістаємо чум, яка теж задовольняє умову мінімальності.

Лінійно упорядковані множини. Чум $(A, \leq)$ називають лінійно упорядкованою або ланцюгом, якщо довільні два її елементи порівнюються відносно $\leq$.

Зрозуміло, що коли лінійно упорядкована множина A має найбільший (найменший) елемент, то цей елемент буде її єдиним максимальним (мінімальним) елементом.

Приклади лінійно упорядкованих множин. 1) Множини $\mathcal{N}, \mathcal{N}^+, \mathcal{Z}, \mathcal{Q}$ і множина $\mathcal{D}$ дійсних чисел з їхнім звичайним порядком. Множину $\mathcal{N}$ часто називають натуральним рядом, а довільну її підмножину виду $\{0, 1, 2, \ldots, n\}$ – початковим відрізком, або просто відрізком натурального ряду.

2) Множина точок числової осі (прямої).

3) Нехай маємо скінченний алфавіт $X = \{x_1, x_2 \ldots, x_n\}$, літери якого лінійно упорядковані: $x_i < x_j \Leftrightarrow i < j$.

Якщо X – алфавіт, то множина всіх слів в алфавіті X позначається як $F(X)$ або X^*. **Лексикографічним порядком** на множині $F(X)$ називають відношення $\leq$, для якого $p = x_{11}x_{12}\cdots x_{1k} \leq x_{21}x_{22}\cdots x_{2r} = q$ тоді і тільки тоді, коли виконується одна із двох умов:

а) слово p є початком слова q;

б) існує таке натуральне число j, що $x_{1j} < x_{2j}$ і для всіх $i < j$ справедливі рівності $x_{1i} = x_{2i}$.

Згідно з лексикографічним порядком упорядковані слова в словниках, якщо прийнятий порядок літер в деякому алфавіті розуміти як лінійний порядок для його літер. ♠

Лінійно упорядковану множину A називають **цілком упорядкованою множиною (цум)**, якщо довільна її непорожна підмножина B має найменший елемент. Звідси випливає, що цум задовольняє умову мінімальності, а тому і двом другим умовам, які їй еквівалентні. У цум для кожного елемента a існує елемент, який безпосередньо слідує за a, за винятком коли a є максимальним елементом. Елемент a може й не мати безпосередньо слідуючого за ним елемента. В такому випадку елемент a називають граничним елементом.

Теорема 2.7 дає можливість застосувати метод математичної індукції не тільки на множині натуральних чисел, а і на довільній фундованій або добре упорядкованій множині. Така можливість приводить до узагальнення методу математичної індукції, який називають методом трансфінітної індукції.

Теорема 2.8 (метод трансфінітної індукції). *Нехай e – найменший* *елемент добре упорядкованої множини A і $P(x)$ – деяка властивість елемента $x \in A$. Тоді, якщо із істинності $P(e)$ і $P(x)$ для всіх $x < a$ випливає істинність $P(a)$, то $P(x)$ істинно для всіх x із A.*

Доведення. Припустимо супротивне, тобто припустимо, що існує така непорожна підмножина A' елементів із A, що $P(a)$ хибне на її елементах при виконанні умов теореми. Нехай a – мінімальний елемент в A'. Оскільки $P(e)$ істинне, то $a \neq e$ і $a > e$. Із умов теореми випливає, що $P(x)$ істинне для всіх $x < a$, але тоді із цих же умов повинна випливати істинність і $P(a)$, а це суперечить нашому припущенню. ■

Метод індукції дає можливість не тільки доводити твердження, але і давати означення за індукцією та виконувати побудову за індукцією. Побудова за індукцією ґрунтується на понятті індуктивного означення.

2.2. Індуктивні означення

Неформально означення за індукцією є способом побудови елементів деякої множини. Цей спосіб полягає в застосуванні певних правил для породження елементів множини таким чином, що елемент належить до даної множини тільки тоді, коли він породжений застосуванням лише цих правил. Формальне означення має вигляд.

Означення 2.6. *а) Правилом називають пару $X \to x$, де X – скінченна множина, елементи якої називають* **посиланнями**, *а елемент x називають* **наслідком**.

б) Нехай Φ – множина правил. Множина A називають Φ-замкнутою, якщо у кожного правила із Φ, посилання якого належать до A, наслідок теж належить до A. Нотація $\Phi : X \to x$ означає, що правило $X \to x$ належить до Φ і множина A буде Φ-замкнутою, якщо $\Phi : X \to x \wedge (X \subseteq A \to x \in A)$.

в) Якщо Φ – множина правил, то множину $I(\Phi)$ називають **індуктивно визначеною** *або* **індуктивною множиною** *за допомогою множини правил Φ тоді і тільки тоді, коли $I(\Phi) = \bigcap\{A \mid A$ є Φ-замкнутою множиною$\}$. Тобто $I(\Phi)$ — це найменша Φ-замкнута множина.*

Наведене означення називають **фінітарним** у зв'язку зі скінченністю множини посилань X.

Це досить абстрактне означення пояснимо трьома прикладами.

1) Множина натуральних чисел $\mathcal{N}$ – це найменша множина, яка включає 0 і замкнута відносно функції слідування $s(x) = x + 1$. Або за означенням: $\Phi_{\mathcal{N}} : \varnothing \to 0$(*базис*), $\{n \in \mathcal{N}\} \to n+1$ (індукція) і $\mathcal{N} = I(\Phi_{\mathcal{N}})$ (замкнутість).

2) Якщо R – бінарне відношення, задане на множині A, то еквівалентне замикання відношення R (тобто рефлексивне, симетричне і транзитивне замикання R) – це найменше відношення еквівалентності, яке визначається за допомогою множини правил: $\varnothing \to (a, a)$, $\varnothing \to (a, b)$, якщо aRb, $\{(a, b)\} \to$ $\to (b, a)$, $\{(a, b), (b, c)\} \to (a, c)$.

3) Нехай $(A, \leq)$ – частково упорядкована множина відношенням $\leq$. **Фундованою множиною** $(A, \leq)$ називається множина $(W, <)$ тих елементів a_i із A, що не існує нескінченно строго спадної послідовності вигляду

$$a_0 > a_1 > a_2 > \dots$$

елементів із A. Індуктивне означення $(W, <)$ має вигляд:

$$\Phi_< : \{x \in A \mid x < a\} \to a \ i \ (W, <) = I(\Phi_<).$$

Множина $(W, <)$ називається фундованою частиною відношення часткового порядку $\leq$.

Індуктивні означення приводять до загального принципу математичної індукції.

Теорема 2.9 (Загальний принцип математичної індукції) . Якщо P – властивість така, що $P(x)$ виконується як тільки $\Phi_< : X \to x$ і $\forall y \in X$ виконується $P(y)$, то $P(a)$ виконується для всіх $a \in I(\Phi_<)$. Цей принцип називається принципом $\Phi_<$-індукції.

Зокрема, нехай S – деяка непорожна множина, $W = I(\Phi_<)$, предикат $P(x) = (x \in S)$ (предикат належності елемента до множини) і $m \in W$ – найменший елемент, тоді, якщо

1) $m \in S$ (базис) і 2) із припущення, що $k \in S$, де $k > m$ (індукція), випливає що і $s(k) \in S$, то $S = W$ (замикання), де $s(x)$ означає елемент, отриманий з елемента x і раніше побудованих елементів множини S за правилами із множини $\Phi_<$. Звідси, як окремий випадок, випливає розглянутий вище узагальнений принцип математичної індукції.

Приклад для ілюстрації. Покажемо як можна застосувати метод математичної індукції на індуктивно визначених множинах.

Мова термів. Алфавітом X цієї мови є символи, розбиті на три групи:

$$T_0 = T_c \cup T_v, \ \Omega \text{ і } \{(,)\}.$$

Символи із T_0 ділять на *предметні константи* $T_c = \{a, b, c, \dots\}$ і *предметні змінні* $T_v = \{x, y, \dots\}$ або ті самі літери з індексами.

Символи із Ω – називають *функціональними.* Це літери з верхніми і, можливо, нижніми індексами: $f_2^3, g_2^5, f_5, \dots$. Верхній індекс n ($n \geq 1$) вказує арність функціонального символу. Якщо його немає, то функціональний символ вважають унарним. І символи третьої групи – це *ліва, права дужки* і *кома.*

Термами називають множину $T(\Omega, X)$, елементи якої мають таке індуктивне означення:

Базис: $T_0 \in T(\Omega, X)$.

Індукція: якщо $t_1, \ldots, t_n \in T(\Omega, X)$ і $f^n \in \Omega$, то $f^n(t_1, \ldots, t_n) \in T(\Omega, X)$. Множину Ω називають *сигнатурою функціональних символів* мови термів. Якщо $t = f(t_1, \ldots, t_n) \in T(\Omega, X)$, то терми $t_1, \ldots, t_n$ називають безпосередніми підтермами терму t.

Мова предикатів першого порядку. Розширимо алфавіт функціональних символів мови термів символами $A_1^{n_1}, A_2^{n_2}, \ldots, A_k^{n_k}, \ldots$, які називають предикатними символами, і символами $\neg, \to, \wedge, \vee, \leftrightarrow, \forall$, які називають логічними зв'язками. Верхній індекс предикатного символу, означає його арність, а символ $\forall$ – називають квантором загальності. Множину предикатних символів називають *сигнатурою предикатів* і позначають її $\mathbf{\Pi}$.

Предикатні символи, коли їх застосувати до термів, породжують елементарні (атомарні) формули, або точніше: якщо A^n – предикатний символ, а $t_1, t_2, \ldots, t_n$ – терми, то $A^n(t_1, t_2, \ldots, t_n)$ – **елементарна формула**.

Мову L предикатів першого порядку визначають індуктивно.

Базис: $A_i^n(t_1, t_2, \ldots, t_n) \in L$ для всіх $i = 1, 2, \ldots$.

Індукція: якщо $A, B \in L$ і $x \in T_v$, то $\neg A,\ A \to B,\ \forall x A \in L$.

Отже, МППП є мовою вигляду $L = (T_0 = T_c \cup T_v, \Omega, \Pi)$.

Інші логічні зв'язки вводяться за допомогою таких тотожностей:

$A \vee B \Leftrightarrow \neg A \to B, \qquad (A \wedge B) \Leftrightarrow \neg(\neg A \vee \neg B) \Leftrightarrow \neg(A \to \neg B),$

$(A \leftrightarrow B) \Leftrightarrow (A \to B) \wedge (B \to A).$

Квантор існування $\exists x A$ визначається двоїстим чином: $\neg \forall x \neg A$. Формула $\forall x A$ читається "для всіх x справедливе A", а формула $\exists x A$ – "існує x такий, що справедливе A".

З наведених індуктивних означень мов видно, що метод математичної індукції можна застосовувати принаймні двома шляхами – за кількістю операцій у термах і за кількістю логічних зв'язок у формулі (тобто, за структурою формули) та за довжиною термів і формули, оскільки і довжина і структура обох цих об'єктів є фундованими множинами.

Рекурсія на фундованих чум. Першим прикладом фундованої множини є множина натуральних чисел.

Другим прикладом фундованої множини є множина $\mathcal{N} \times \mathcal{N}$ (меншою є та пара, у якої другий компонент менший, а коли вони рівні, то меншою є пара у якої перший компонент менший). Загальнішу властивість фундованих множин дає

Теорема 2.10 *Якщо A і B – фундовані чум, то їхній декартів добуток $A \times B$ теж фундована множина за відношенням* $(a_1, b_1) \le (a_2, b_2) \Leftrightarrow (b_1 \le b_2) \vee (b_1 = b_2 \wedge a_1 \le a_2)$.

Доведення. Спочатку стабілізуються другі пари, а потім стабілізуються перші пари. ■

Звідси випливає, що фундованими будуть множини $\mathcal{N} \times \mathcal{N} \times \mathcal{N}$ і $\mathcal{N}^n$.

Ще простіше доводиться твердження, що об'єднання двох фундованих множин A і B без спільних елементів, буде фундованою множиною. Дійсно, спадна послідовність $a_1 > a_2 > a_3 > \ldots$ або повністю належить множині A або повністю належить множині B.

Принцип $\Phi_<$-індукції у випадку фундованості відношення $\leq$ стає принципом трансфінітної індукції за фундованим відношенням $\leq$. А з трансфінітною індукцією пов'язаний метод побудови функцій за трансфінітною рекурсією. А зв'язок фундованості і доброї упорядкованості очевидний: *фундована лінійно упорядкована множина є добре упорядкованою множиною.*

Рекурсією називають такий спосіб означення функції, за яким значення функції для довільних значень аргументів відомим чином виражаються через значення для менших значень аргументів. Іншими словами, цей спосіб дає можливість визначити єдину функцію f на множині $(W, <)$ таку, що для $a \in (W, <)$ значення $f(a)$ відомим чином виражається через $f(x)$ для $x < a$.

Нехай A – добре упорядкована множина і на цій множині визначена функція $f(x)$, яка ставить у відповідність кожному елементу x із A деякий елемент множини B. Припустимо також, що $f(x)$ повинна задовольняти деяким *рекурентним співвідношенням* (які також називають **рекурсивною залежністю**), тобто співвідношенням, які однозначно визначають для довільного $a \in A$ значення $f(a)$ за значеннями $f(b)$ для всіх $b < a$.

Метод побудови функції за рекурсією ґрунтується на теоремі 2.11.

Теорема 2.11. *Існує єдина функція $f(x)$, яка визначена на всій множині A, задовольняє вказаним рекурентним співвідношенням і приймає задані значення на мінімальному елементі множини A.*

Доведення. Покажемо спочатку єдиність такої функції. Припустимо, що існує дві різні функції $f(x)$ і $g(x)$ на множині A, які задовольняють нашим умовам. Нехай існує непуста підмножина елементів x із A, для яких $f(x) \neq g(x)$. Оскільки A – добре упорядкована, то ця підмножина має мінімальний елемент a. Цей елемент не може бути мінімальним для всієї множини A, тому що тоді, за умовою, на цьому елементі $f(a)$ і $g(a)$ збігалися б. Отже, існує $b < a$ такий, що $f(b) = g(b)$. За умовою теореми, рекурентні співвідношення однозначно визначають значення наших функцій для $x = a$ по їх значеннях для всіх $b < a$, а це означає, що $f(a) = g(a)$. Одержана суперечність доводить єдиність $f(x)$.

Доведемо тепер існування функції. Припустимо, що на мінімальному елементі множини A значення шуканої функції уже задано. Позначимо через P таку властивість: елемент $a \in A$ задовольняє властивості P, якщо на множині C всіх таких x, що $x \leq a$, може бути визначена функція $f_a(x)$, яка задовольняє рекурентним співвідношенням і приймає задане значення на мінімальному елементі множини A.

В силу припущення, P істинне на мінімальному елементі $e \in A$. Далі, якщо елементи b і c задовольняють властивості P і $b < a$, то на підставі доведеної вище єдиності шуканої функції не на множині A, а на множині $B = \{x \in A \mid x \leq b\}$, маємо $f_b(x) = f_a(x)$.

Звідси випливає, що коли всі елементи b, строго менші елемента a, задовольняють властивості P, то і сам елемент a задовольняє цій властивості. Одержуємо функцію $f_a(x)$, яка задовольняє всі вимоги, якщо для довільного елемента $b < a$ покласти $f_a(b) = f_b(b)$, а за $f_a(a)$ взяти те значення, яке однозначно визначається рекурентними співвідношеннями.

На основі методу трансфінітної індукції можна говорити, що для всіх a із A істинно $P(a)$. Покладаючи тепер $(\forall a \in A)\ f_a(a) = f(a)$, визначаємо функцію $f(x)$, яка має всі необхідні властивості. ■

Приклад 1.2.23. Розглянемо приклади індуктивного та рекурсивного способу означення.

1) (*Індуктивне означення*). Нехай $X = \{x_1, x_2, \ldots, x_n\}$ – скінченний алфавіт. Слово p в алфавіті X називається *паліндромом*, якщо воно читається зліва направо і справа наліво однаково (наприклад, слово "*пилип*"). Це неформальне поняття паліндрома має таке індуктивне означення.

а) пусте слово e – паліндром; б) якщо $x \in X$, то x – паліндром;
в) якщо слово p – паліндром, то xpx – паліндром для всіх $x \in X$;
г) паліндромами є ті і тільки ті слова, які побудовані за правилами а) – в).

2) (*Рекурсивне означення*). а) Нехай функція $f(n)$, де $n \in \mathcal{N}$, задана таким рекурентним співвідношенням:

$$T(n) = \begin{cases} b, & \text{якщо} \quad n = 0; \\ rT(n-1) + a, & \text{якщо} \quad n > 0, \end{cases}$$

де b, r – деякі константи (не обов'язково з множини $\mathcal{N}$), причому $r \neq 1$.

Виконуючи очевидні підстановки і обчислення, отримуємо

$$T(n) = r(rT(n-2) + a) + a = \ldots = r^n T(0) + a \sum_{i=0}^{n-1} r^i = br^n + a\frac{1-r^n}{1-r},$$

якщо $r < 1$ і $T(n) = br^n + a\frac{r^n-1}{r-1}$, якщо $r > 1$. Отже, $f(n) = br^n + a\frac{1-r^n}{1-r}$, якщо $r \neq 1$. Надаючи значення параметрам a, b, r, дістаємо

а1) при $a = 1, b = 0, r \neq 1$ суму n перших членів геометричної прогресії, тобто $f(n) = \sum_{i=0}^{n-1} r^i = \frac{1-r^n}{1-r}$;

а2) при $a = 1, r = 2, b = 0$ рекурентне співвідношення приймає вигляд

$$T(n) = \begin{cases} 0, & \text{якщо} \quad n = 0; \\ 2T(n-1) + 1, & \text{якщо} \quad n > 0, \end{cases}$$

а функція $f(n) = 2^n - 1$;

а3) при $a = 0, r = 2, b = 1$ рекурентне співвідношення приймає вигляд

$$T(n) = \begin{cases} 1, & \text{якщо} \quad n = 0; \\ 2T(n-1), & \text{якщо} \quad n > 0, \end{cases}$$

а $f(n) = 2^n$.

Рекурентність і математична індукція ідеально підходять одне до одного. Прикладом можуть служити задачі 1.1.8, 1.2.4, 1.4.2 з попереднього підрозділу.

Далі ми ще повернемося до визначення за індукцією та рекурсією.

Розділ 3

ІНДУКТИВНІ МОДЕЛІ

Неформально під словом модель розуміють систему, яка зображує на деякому рівні абстракції реальний об'єкт і зв'язки між складовими цього об'єкта. А з математичної точки зору (тобто, з формального погляду) модель складається з деякої непорожної множини та множин операцій і предикатів, визначених на цій множині.

Визначившись з основою методу математичної індукції, розглянемо індуктивні моделі та означення операцій на таких моделях. Розглянемо лише моделі Пеано та індуктивні моделі, які з ними пов'язані, оскільки такі означення аналогічним чином вводяться і на інших індуктивних моделях.

3.1. Алгебричні системи

Поняття алгебричної системи поєднує у собі поняття алгебри і логічного числення. Формальним апаратом теорії алгебричних систем є числення предикатів першого порядку, а сама теорія може розглядатися як теорія, що міститься між математичною логікою і алгеброю.

Означення 3.1. *Алгебричною системою* (AC) *називають упорядковану трійку* $\mathcal{A} = (A, \Omega, \Pi)$, *де* A *– деяка непорожна множина,* $\Omega = \{\omega_1^{n_1}, \omega_2^{n_2}, \ldots\}$ *— множина операцій (сигнатура операцій) і* $\Pi = \{\pi_1^{k_1}, \pi_2^{k_2}, \ldots\}$ *— множина предикатів (сигнатура предикатів), визначених на множині* A.

Типом AC *називають пару множин* $\tau = (\{n_1, n_2, \ldots\}, \{k_1, k_2, \ldots\})$, *елементами яких служать відповідно арності операцій і предикатів. Тип алгебричної системи* τ *скінченний, якщо скінченні обидві його множини.*

Множину A *називають* **носієм** *або* **основною множиною системи** $\mathcal{A} = (A, \Omega, \Pi)$, *а її елементи — елементами системи* $\mathcal{A}$.

Потужність $|A|$ *множини* A *називають* **потужністю** *або* **порядком** AC $\mathcal{A} = (A, \Omega, \Pi)$.

На відміну від інших операцій і предикатів, які можуть бути визначені на множині A, операції з Ω і предикати з Π називають **основними**. Значення основних нульарних операцій АС називають головними або **виділеними елементами** цієї системи.

Алгебричні системи $\mathcal{A} = (A, \Omega, \Pi)$ і $\mathcal{B} = (B, \Omega', \Pi')$ називають алгебричними системами того самого типу, якщо між множинами їх операцій і предикатів існує така взаємно однозначна відповідність, за якої відповідні операції і предикати мають однакові арності. Відповідні операції називають операціями **того самого типу** або **однотипними**. Часто однотипні операції і предикати не розрізняють між собою і вважають, що системи $\mathcal{A} = (A, \Omega, \Pi)$ і $\mathcal{B} = (B, \Omega', \Pi')$ мають однакові сигнатури операцій і предикатів.

АС скінченного типу $\mathcal{A} = (A, \Omega, \Pi)$ називають **скінченною**, якщо множина A скінченна. У цьому випадку АС $\mathcal{A} = (A,\ \Omega, \Pi)$ записують $\mathcal{A} = (A; \omega_1^{n_1}, \omega_2^{n_2}, \ldots, \omega_s^{n_s}; \pi_1^{k_1},\ \pi_2^{k_2}, \ldots, \pi_r^{k_r})$.

АС $\mathcal{A} = (A, \Omega, \Pi)$ називають **одноосновною алгеброю**, якщо $\Pi = \emptyset$, і **моделлю**, якщо $\Omega = \emptyset$.

Приклад 3.3.1. $\mathcal{A} = (\mathcal{Z}; \{+\}), \mathcal{A} = (\mathcal{Q}; \{+, -\}), \mathcal{A} = (\mathcal{Z}; \{+\}; \{\leq\}), \mathcal{A} = (Z; \{\leq\}),\ \ \mathcal{A} = (\mathcal{N}; \{s, +, 0, 1\})$, де $\mathcal{Z}, \mathcal{Q}, \mathcal{N}$ – множини всіх цілих, раціональних і натуральних чисел відповідно, а "+" і "–" — звичайні операції додавання і віднімання чисел. Предикат $\leq$ – це відношення лі- нійного порядку. Перша і друга АС — алгебри типу (2) і (2, 2), третя АС – типу (2; 2), четверта — модель типу (2), а п'ята – алгебра типу (1, 2, 0, 0), де $s(x) = x + 1$, а 0 і 1 – виділені елементи цієї АС.

Мова предикатів першого порядку, яка була визначена вище, є алгебричною системою вигляду (A, Ω, Π). ♠

Якщо в означенні АС замінити слово операцій словами часткових операцій, то отримуємо означення часткової АС.

Довільній n-арній операції на множині A відповідає $(n + 1)$-арний предикат P_w. Дійсно,

$$P_w(a_1, a_2, \ldots, a_n, b) = 1 \Leftrightarrow w(a_1, a_2, \ldots, a_n) = b,$$

де $a_1, a_2, \ldots, a_n, b \in A$.

Якщо таким способом замінити в АС усі операції предикатами, то дістаємо модель $\mathcal{A}^* = (A, \Pi^*)$, яку називають моделлю, що відповідає АС $\mathcal{A} = (A, \Omega, \Pi)$.

Приклад 3.3.2. Для $\mathcal{A} = (Z; \{+, -, 0\}; \{\leq\})$ модель набуває вигляду $\mathcal{A}^* = (Z;\ \{P_+^3, P_-^3, P_0^1, \leq\})$, де

$P_+(x, y, z) = 1 \Leftrightarrow x + y = z,\ \ P_-(x, y, z) = 1 \Leftrightarrow x - y = z,\ P_0(x) = 1 \Leftrightarrow x = 0.$ ♠

Відображенням АС $\mathcal{A} = (A, \Omega, \Pi)$ в АС $\mathcal{B} = (B, \Omega', \Pi')$ називають відображення $h : A \to B$.

Означення 3.2. *Ізоморфізмом АС* $\mathcal{A} = (A, \Omega, \Pi)$ *типу* τ *в АС* $\mathcal{B} = (B, \Omega', \Pi'))$ *того самого типу* τ *називають взаємно однозначне відображення* h*, яке задовольняє такі умови:*

$$(i)\ h(\omega(a_1, a_2, \ldots, a_n)) = \omega'(h(a_1), h(a_2), \ldots, h(a_n)),$$
$$(i1)\ \pi_k(a_1, a_2, \ldots, a_n) \Leftrightarrow \pi'_k(h(a_1), h(a_2), \ldots, h(a_n)),$$

для будь-яких $a_1, a_2, \ldots, a_n$ *з* A *і для відповідних* ω *із* Ω *і* ω' *із* Ω' *та* π_k *з* Π *і* π'_k *з* Π'.

Ізоморфізм $h : \mathcal{A} \to \mathcal{A}$ *називають* **автоморфізмом**.

Гомоморфізмом АС $\mathcal{A} = (A, \Omega, \Pi)$ *в АС* $\mathcal{B} = (B, \Omega', \Pi')$ *того самого типу, що й* $\mathcal{A}$*, називають відображення* $h : A \to B$*, яке задовольняє умову* (i) *й умову*

$$(i2)\ \pi_k(a_1, a_2, \ldots, a_n) \to \pi'_k(h(a_1), h(a_2), \ldots, h(a_n)),$$

для всіх $a_1, a_2, \ldots, a_n \in A,\ \pi_k \in \Pi$ *і* $\pi'_k \in \Pi'$.

Із цих означень випливає, що довільний ізоморфізм АС є гомоморфізмом. Взаємно однозначний гомоморфізм для АС не завжди буде ізоморфізмом. Дійсно, нехай $\mathcal{A} = (\mathcal{N}, \pi)$ і $\mathcal{B} = (\mathcal{N}, <)$, де π – бінарне тотожно хибне відношення на $\mathcal{N}$. Моделі $\mathcal{A}$ і $\mathcal{B}$ одного типу, а довільне відображення $\mathcal{A}$ в $\mathcal{B}$ буде гомоморфізмом. Справді, умова $(i2)$ тут завжди виконується, а умову (i) і перевіряти не потрібно, оскільки немає операцій. Але між $\mathcal{A}$ і $\mathcal{B}$ немає жодного ізоморфізму. Якби це було не так, тобто, якщо h – ізоморфізм, то $h(0)$ і $h(1)$ різні і, отже, або $h(0) < h(1)$, або $h(1) < h(0)$. З умови $(i1)$ випливає, що $\pi(0, 1)$ або $\pi(1, 0)$ буде істинним, що суперечить тому, що π тотожно хибне на $\mathcal{N}$.

Гомоморфізм h АС $\mathcal{A} = (A, \Omega, \Pi)$ в АС $\mathcal{B} = (B, \Omega', \Pi')$ називають гомоморфізмом "на", якщо h – відображення A на B.

З означень гомоморфізму й ізоморфізму випливає

Теорема 3.1. *1. Довільний взаємно однозначний гомоморфізм* h *скінченної АС* $\mathcal{A} = (A, \Omega, \Pi)$ *на себе є ізоморфізмом.*

2. Добуток гомоморфізмів (ізоморфізмів) АС є гомоморфізмом (ізоморфізмом) АС.

3. Відношення ізоморфності АС є відношенням еквівалентності.

Доведення пропонуються читачеві як прості вправи.

Теорема 3.2. *Відображення h $\mathcal{A} = (A, \Omega, \Pi)$ в $\mathcal{B} = (B, \Omega', \Pi')$ є гомоморфізмом тоді і тільки тоді, коли h — гомоморфізм моделі $\mathcal{A}^*$ в модель $\mathcal{B}^*$, де $\mathcal{A}^*$ і $\mathcal{B}^*$ – моделі, які відповідають АС $\mathcal{A}$ і $\mathcal{B}$.*

Доведення. Нехай h – гомоморфізм $\mathcal{A}^*$ в $\mathcal{B}^*$, а ω і ω' – однотипні основні операції АС $\mathcal{A}$ і $\mathcal{B}$, а P_w і $P_{w'}$ – відповідні предикати моделей $\mathcal{A}^*$ і $\mathcal{B}^*$. Нехай $a_1, \ldots, a_n \in A$. Тоді покладемо $\omega(a_1, \ldots, a_n) = a$. Отримуємо $P_w(a_1, \ldots, a_n, a) = 1$. Отже, $P_{w'}(h(a_1), \ldots, h(a_n), h(a)) = 1$, тобто $\omega'(h(a_1), \ldots, h(a_n)) = h(a)$.

Доведення у зворотному напрямку виконують аналогічно. ■

АС $\mathcal{A}_1 = (A_1, \Omega, \Pi)$ називають підсистемою АС $\mathcal{A} = (A, \Omega, \Pi)$, якщо $A_1 \subseteq A$ і множина A_1 замкнута відносно будь-якої основної операції із Ω, а значення основних предикатів із Π на множині A_1 збігаються із значеннями тих же предикатів на A для всіх елементів з A_1. Якщо $\mathcal{A}$ – алгебра або модель, то $\mathcal{A}_1$ називають **підалгеброю** або **підмоделлю** $\mathcal{A}$. Зокрема, якщо $\mathcal{A}$ являє собою модель, то довільна непорожна підмножина носія буде замкнутою і, отже, буде підмоделлю моделі $\mathcal{A}$. Для АС справедлива

Теорема 3.3. *Непорожній перетин довільної сукупності підсистем АС $\mathcal{A} = (A, \Omega, \Pi)$ є підсистемою АС $\mathcal{A}$.*

3.2. Індуктивні моделі на $\mathcal{N}$

У зв'язку з тим, що метод математичної індукції тісно пов'язаний з множиною натуральних чисел $\mathcal{N}$, то розглянемо детальніше моделі, які виникають у цій множині та їхні властивості.

Далі під словом модель розумітимемо АС вигляду $M = (N; 0, s; =)$, де N – деяка множина; $0, s$ – відповідно нульарна і унарна операції та $=$ – бінарний предикат рівності, визначені на N. Таку модель називають **моделлю Пеано**, якщо вона задовольняє такі три аксіоми (умови) [10]:

P1. Для всіх $x \in N$ $s(x) \neq 0$.

P2. Для всіх $x, y \in N$, якщо $x \neq y$, то $s(x) \neq s(y)$.

P3. Якщо G підмножина N така, що

а) $0 \in G$ і б) якщо $x \in G$ і $s(x) \in G$, то $G = N$.

Якщо N^* – множина натуральних чисел у тому вигляді, в якому їх сприймає наша інтуїція, то 0^* – це нуль, s^* – це операція "слідує за" (тобто, це така операція, що для довільного x $s^*(x)$ є наступним за x натуральним числом) і $\mathcal{N}^* = (N^*; 0^*, s^*; =)$ – приклад моделі Пеано. Саме цей приклад моделі натуральних чисел сприяв розгляду вищенаведених аксіом, які називають аксіомами Пеано.

Зауважимо, що моделей Пеано існує багато. Прикладом такої моделі є модель парних натуральних чисел $(N'; 0', s'; =)$, де $0'$ – це число 2, $s'(x)$ – наступне парне число за числом x.

Аксіому P3 називають аксіомою індукції.

Означення 3.3. *Модель називають індуктивною, якщо вона задовольняє аксіому математичної індукції.*

Отже, довільна модель Пеано є індуктивною моделлю, але обернене твердження не виконується. Покажемо це на прикладах.

Приклад 3.3.3. Нехай N'' – множина, яка включає два елементи a_0 і a_1. Нехай унарна операція s'' на N'' має стале значення a_1, тобто $s''(x) = a_1$ для всіх $x \in N''$. Тоді $M'' = (N''; a_0, s''; =)$ є індуктивною моделлю, але не буде моделлю Пеано, оскільки вона не задовольняє аксіому P2.

Приклад 3.3.4. Другим прикладом індуктивної моделі, в якій не виконується аксіома P1, є модель $M''' = (N'''; 0''', s'''; =)$, де $N''' = \{0'''\}$, а $s'''(0''') = 0'''$. Отже, ця модель не є моделлю Пеано.

Виникає питання: чи існують індуктивні моделі, які не задовольняють аксіоми P1 і P2? Виявляється, що таких моделей не існує: кожна модель, яка задовольняє аксіому P3, повинна задовольняти або аксіому P1, або аксіому P2.

Доведення цього факту буде дано далі після побудови теорії визначення за індукцією.

З аксіом P1–P3 випливають такі властивості:

P4. Якщо $y \in N$ і для всіх $x \in N$ $y \neq s(x)$, то $y = 0$.

P5. Для всіх $x \in N$ $s(x) \neq x$.

Справедливість цих властивостей для моделі Пеано $\mathcal{N}^* = (N^*; 0^*, s^*; =)$ очевидна. P4 означає, що число, яке не слідує ні за яким іншим числом, є 0, а P5 означає, що жодне із чисел не слідує за самим собою.

Але існує важлива різниця між P4 і P5. Доведення P4 використовує лише аксіому P3 і тому P4 справедлива для всіх індуктивних моделей, а доведення P5 використовує всі три аксіоми P1, P2 і P3. Приклад моделі $M''' = (N'''; 0''', s'''; =)$ показує, що P5 справедлива не для всіх індуктивних моделей.

3.3. Означення за індукцією

Розширимо наші початкові поняття "число", "нуль", "слідує за" такими поняттями як операції *додавання, множення, піднесення до степеня* тощо.

Спочатку розглянемо операцію додавання і спосіб її визначення.

Додавання – це бінарна операція, визначена на множині натуральних чисел, тобто – це функція "$+$" двох змінних, яка кожній упорядкованій парі (x, y) натуральних чисел ставить у відповідність нове натуральне число, яке позначається як $x + y$. Пеано визначає $x + y$ двома умовами:

$$x + 0 = x, \tag{1.1}$$
$$x + s(y) = s(x + y). \tag{1.2}$$

Інтуїтивні представлення про операцію додавання переконують нас, що ці умови виконуються для всіх натуральних чисел x і y.

Але в якому сенсі умови (1.1) і (1.2) складають *означення* операції додавання? Зокрема, чи визначається деяка бінарна операція, що задовольняє ці умови, однозначно і чи це виконується в довільній моделі Пеано?

Відповіді на ці запитання ми розглядали для фундованих множин і добре упорядкованих множин, а чи будуть результати, отримані для цих множин, справедливими для індуктивних моделей і моделей Пеано?

Для відповіді на ці питання розглянемо одну, пов'язану з ними, проблему.

Введення операції за допомогою умов (1.1) і (1.2) є прикладом **визначення за індукцією**. Для розгляду цього поняття в загальних термінах, задамо модель Пеано $M = (N; 0, s; =)$ і другу модель $M_1 = (N_1; 0_1, s_1; =)$, яка не обов'язково повинна бути моделлю Пеано (або, навіть, індуктивною моделлю). Будемо говорити, що дві рівності

$$h(0) = 0, \tag{2.1}$$
$$h(s(y)) = s_1(h(y)). \tag{2.2}$$

визначають за індукцією функцію h, яка відображає N в N_1 (тобто $h : N \to N_1$), якщо для всіх $y \in N$ виконуються рівності (2.1) і (2.2). Функцію h, яка задовольняє умови (2.1) і (2.2) можна назвати гомоморфізмом $\mathcal{N} = (N; 0, s; =)$ в $M_1 = (N_1; 0_1, s_1; =)$. Знову виникає питання: в якому сенсі ці умови визначають функцію h. Відповідь на це питання дає

Теорема 3.4. *Для довільної моделі Пеано $M = (N; 0, s; =)$ і довільної моделі $M_1 = (N_1; 0_1, s_1; =)$ існує одна і тільки одна функція h, яка відображає M в M_1 та задовольняє умови (2.1) і (2.2).*

Перш ніж доводити цю теорему, з'ясуємо її зв'язок з умовами (1.1) і (1.2).

Нехай $M = (N; 0, s; =)$ – довільна модель Пеано, тоді для довільного $x \in N$ побудуємо модель Пеано $M_x = (N_x; x, s; =)$, де N_x – підмножина множини N, елементами якої є $x, s(x), s(s(x)), \ldots$. Застосовуючи теорему 3.4

до моделей M і M_x, дістаємо, що для кожного $x \in N$ існує єдина функція h_x, яка відображає M у себе так, що виконуються умови

$$h_x(0) = x, \tag{3.1}$$
$$h_x(s(y)) = s(h_x(y)) \tag{3.2}$$

для всіх $y \in N$. З існування та єдиності такої функції h_x і виводиться існування єдиної бінарної операції додавання на моделі $M = (N; 0, s; =)$, причому цей висновок ґрунтується тільки на теоретико-множинних побудовах і не залежить від аксіом P1–P3.

Нехай f – бінарна операція на N така, що $\forall x, y \in N$ існує елемент $f(x, y)$, який визначається таким способом:

$$f(x, y) = h_x(y). \tag{4}$$

На підставі (4), (3.1), (3.2) отримуємо, що f задовольняє умови (для всіх $x, y \in N$):

$$h_x(0) = x, \tag{5.1}$$
$$h_x(s(y)) = s(f(x, y)). \tag{5.2}$$

Далі f – єдина бінарна операція на N, яка задовольняє ці умови. Дійсно, нехай g – деяка бінарна операція на N, яка задовольняє умови

$$g(x, 0) = x, \tag{6.1}$$
$$g(x, s(y)) = s(g(x, y)) \tag{6.2}$$

для всіх $x, y \in N$. Поставимо у відповідність кожному $x \in N$ унарну операцію g_x на N таку, що для всіх $y \in N$

$$g_x(y) = g(x, y). \tag{7}$$

Із (7), (6.1), (6.2) випливає, що для всіх $x \in N$ справедливі рівності

$$g_x(0) = x, \tag{8.1}$$
$$g_x(s(y)) = s(g_x(y)). \tag{8.2}$$

Якщо порівняти (3.1), (3.2) і (8.1), (8.2), то стає очевидним, що $g_x = h_x$ для кожного $x \in N$, оскільки з теореми 3.4 випливає єдиність функції h_x, яка визначається умовами (3.1) і (3.2). Але на підставі (4) і (7) дістаємо $f(x, y) = g(x, y)$ для всіх $x, y \in N$. А це й означає, що операції f і g на N збігаються. Отже, з теореми 3.4 випливає, що для довільної моделі Пеано $\mathcal{N} = (N; 0, s; =)$ існує єдина бінарна операція f на N, яка задовольняє умови (5.1) і (5.2) для всіх $x, y \in N$. Цю бінарну операцію і називають **додаванням**, а коли її позначити символом $+$, то умови (1.1) і (1.2) – це умови (5.1) і (5.2) тільки записані іншим способом.

Крім того, зі сказаного випливає також те, що однієї аксіоми математиченої індукції недостатньо для означення за індукцією (потрібна принаймні аксіома P1).

Доведення теореми 3.4. Нехай $M = (N; 0, s; =)$ – довільна модель Пеано і $M_1 = (N_1; 0_1, s_1; =)$ – довільна інша модель. Підмножину H множини N називають **сегментом**, якщо

$$1)\ 0 \in H \text{ і } 2) \text{ як тільки } s(x) \in H, \text{ то } x \in H.$$

Визначимо часткову функцію j таку, що

1) вона визначена на деякому сегменті H і
2) набуває значень, які належать множині N_1, причому умови

$$j(0) = 0_1, \tag{10.1}$$
$$j(s(x)) = s_1(j(x)) \tag{10.2}$$

виконуються для всіх x таких, що $s(x) \in H$.

Лема 3.1. *Кожний елемент з N належить області визначення якоїнебудь часткової функції, що була визначена у абзаці вище. , що була визначена у абзаці вище*

Доведення. Нехай G – множина елементів з N, які належать області визначення якої-небудь часткової функції. Одноелементну множину $\{0\}$ можна вважати сегментом, оскільки на підставі аксіоми P1, не існує такого x, що $s(x) \in \{0\}$. Із цих причин функція j з областю визначення $\{0\}$ і значенням $j(0) = 0$, буде частковою функцією. Отже, $0 \in G$.

Припустимо, що y – довільний елемент із G і j – часткова функція, область визначення якої сегмент H, і H включає елемент y. Якщо $s(y) \in H$, то $s(y)$ також належить G. Тому розглянемо випадок, коли $s(y) \notin H$. Нехай H' означає підмножину множини N, яка отримана шляхом додавання $s(y)$ до H, і нехай j' – функція, область визначення якої є H', і значення якої задають таким способом.

Якщо $x \in H$, то $j'(x) = j(x)$, і якщо $x = s(y)$, то $j'(x) = s_1(j(y))$. Далі буде показано, що j' – часткова функція, і тому коли $s(y) \notin H$ (так само, як у протилежному випадку, про який йшлося вище), то $s(y) \in G$. Оскільки множина G включає 0 і замкнута відносно s, то на підставі аксіоми P3 маємо $G = N$.

Тому, щоб закінчити доведення леми, залишається показати, що j' є часткова функція. Із цією метою розширимо спочатку область визначення H' часткової функції j', яка включає множину H і елемент $\{s(y)\}$. Оскільки H – область визначення часткової функції, то вона є сегментом і тому $0 \in$

$\in H'$. Далі, якщо x – довільний елемент із N такий, що $s(x) \in H'$, то також і $x \in H'$ на підставі того, що а) коли $s(x) \in H$, то $x \in H$ на підставі того, що H – сегмент; б) якщо $s(x) = s(y)$, то $x = y$ на підставі аксіоми P2, і знову $x \in H$ (оскільки відомо, що $y \in H$). Ці міркування показують, що H' *є сегментом.*

Далі, $j'(0) = j(0) = 0_1$ і $j'(s(x)) = s_1(j'(x))$ як тільки $s(x) \in H'$. Остання рівність ґрунтується на розгляді окремо тих самих двох випадків, що і вище:

а) якщо $s(x) \in H$, то $x \in H$, так що

$$j'(s(x)) = j(s(x)) = s_1(j(x)) = s_1(j'(x));$$

б) якщо $s(x) = s(y)$, то $x = y$ і

$$j'(s(y)) = s_1(j(y)) = s_1(j'(y)).$$

Оскільки область визначення функції j є сегментом і j' задовольняє умови (10.1) і (10.2), то j' – часткова функція. ■

Лема 3.2. *Якщо j_1 і j_2 – дві часткові функції і x належить області визначення кожної із цих функцій, то $j_1(x) = j_2(x)$.*

Доведення. Нехай G — множина таких елементів $x \in N$, що має місце рівність $j_1(x) = j_2(x)$ як тільки j_1 і j_2 часткові функції, області визначення яких включають елемент x. Очевидно, що $0 \in G$, оскільки $j(0) = 0_1$ для кожної часткової функції j.

Нехай x – довільний елемент із G, а j_1 і j_2 – довільні часткові функції, область визначення кожної з них включає $s(x)$. Тоді $j_1(s(x)) = s_1(j_1(x))$ і $j_2(s(x)) = s_1(j_2(x))$. Але оскільки $x \in G$, то $j_1(x) = j_2(x)$ і $j_1(s(x)) = j_2(s(x))$. Тому $s(x) \in G$. Оскільки $0 \in G$ і G замкнута відносно s, то на піставі аксіоми P3 дістаємо $G = N$. ■

З лем 3.1 і 3.2 випливає, що для довільного $x \in N$ існує один і тільки один елемент $z \in N_1$ такий, що $z = j(x)$ для довільної часткової функції j, область визначення якої включає елемент x (принаймні одна така функція знайдеться). Нехай h – функція з областю визначення N така, що $\forall x \in N$ $h(x)$ визначається саме цим елементом $z \in N_1$. Виявляється, що ця функція h задовольняє умови (2.1) і (2.2) і що вона єдина функція з областю визначення N, яка має цю властивість.

Дійсно, $h(0) = j(0) = 0_1$ для довільної часткової функції j. Далі, для довільного $y \in N$ існує така часткова функція j, що $s(y)$ належить області визначення j на підставі леми 3.1. Звідси дістаємо $h(s(y)) = j(s(y)) = s_1(j(y)) = s_1(h(y))$, так, що коли h задовольняє умови (2.1) і (2.2) для всіх $y \in N$, то $h_1 = h$. Це випливає з того, що N є сегментом, де обидві функції є частковими, і тоді на підставі леми 3.2 $h(x) = h_1(x)$ для всіх $x \in N$. А це й означає рівність функцій h і h_1. Цим доведення теореми 3.4 закінчується. ■

3.4. Операції на індуктивних моделях

З теореми 3.4 випливає існування в кожній моделі Пеано єдиної бінарної операції додавання, тобто операції f, яка задовольняє умови (5.1) і (5.2) для всіх $x, y \in N$. Але є загальніший результат.

Теорема 3.5. *В кожній індуктивній моделі існує лише одна операція додавання.*

Доведення цієї теореми не може опиратися на теорему 3.4, оскільки теорема 3.4 справедлива не у всіх індуктивних моделях (напр., в розглянутому вище прикладі 3.3.4 модель $M'' = (N''; 0'', s''; =)$). Для доведення теореми скористаємося такою лемою.

Лема 3.3. *Якщо* $M = (N; 0, s; =)$ *– довільна індуктивна модель, то для кожного елемента* $x \in N$ *існує єдина унарна операція* h_x *на* N *така, що умови (3.1) і (3.2) виконуються для всіх* $y \in N$.

Доведення. Зауважимо, що для довільного $x \in N$ може існувати щонайбільше одна операція h_x, яка задовольняє умови (3.1) і (3.2). Дійсно, припустимо, що h_x і h'_x – дві операції, які задовольняють ці умови, і припустимо, що G така підмножина множини N, до якої входить y тоді і тільки тоді, коли $h_x(y) = h'_x(y)$. Очевидно, що $0 \in G$, оскільки $h_x(0) = h'_x(0)$. Крім того, множина G замкнута відносно операції s, тоді, якщо $y \in G$ (а це тоді, коли $h_x(y) = h'_x(y)$), то $h(s(y)) = s(h(x)) = s(h'_x(y)) = h'_x(s(y))$, а звідси випливає, що $s(y) \in G$. На підставі справедливості аксіоми P3 для моделі $M = (N; 0, s; =)$ дістаємо, що $G = N$, і тому $h_x = h'_x$.

Нехай тепер H – підмножина множини N, яка включає тільки ті елементи, для яких існує операція h_x. Позначимо h_0 тотожну операцію на N, тобто $h_0(y) = y$ для всіх $y \in N$. Звідси отримуємо $h_0(0) = 0$ (тобто виконується умова (3.1)). Отже, $0 \in H$. Далі, множина H замкнута відносно s. Покладемо $x \in H$ і що операція h_x існує. Нехай $h_{s(x)}$ така операція на N, що $h_{s(x)}(y) = s(h_x(y))$ для всіх $y \in N$, тоді $h_{s(x)}(0) = s(h_x(0)) = s(x)$ (отже, для $s(x)$ виконується умова (3.1)), в той час як для довільного $y \in N$

$$h_{s(x)}(s(y)) = s(h_x(s(y))) = s(s(h_x(y)) = s(h_{s(x)}(y)).$$

Отже, для $s(x)$ виконується умова (3.2), а звідси випливає, що $s(x) \in H$. Застосовуючи аксіому P3, яка за умовою справедлива для $M = (N; 0, s; =)$, дістаємо справедливість рівності $H = N$, а це доводить лему. ■

Ця лема дає можливість закінчити доведення теореми 3.5 точно так само, як це було зроблено для доведення існування операції додавання в теоремі 3.4.

Операція множення. В розгляді операції множення ситуація повністю аналогічна тій, яка була в розгляді операції додавання. Під операцією множення для індуктивної моделі $M = (N; 0, s; =)$ розуміють таку бінарну операцію $\cdot$ на N, що для всіх $x, y \in N$

$$x \cdot 0 = 0, \tag{11.1}$$
$$x \cdot s(y) = (x \cdot y) + x, \tag{11.2}$$

де $+$ означає операцію додавання для цієї моделі.

Використовуючи доведення, аналогічне наведеному вище для теореми 3.5, можна показати, що в індуктивній моделі існує одна і тільки одна операція множення.

Оскільки операції додавання і множення єдині в індуктивних моделях, а тому і в моделях Пеано, то це дає можливість визначати інші функції і операції, які опираються на ці операції. Але під час визначення таким способом функцій і операцій слід пам'ятати про модель, де ці об'єкти визначаються. Річ у тім, що модель Пеано індуктивна, але не кожна індуктивна модель буде моделлю Пеано, як про це свідчить модель прикладу 3.3.4. Така ситуація виникає в означенні операції піднесення до степеня.

Під операцією піднесення до степеня для індуктивної моделі $(N; 0, s; =)$ розуміють таку бінарну операцію exp, що для всіх $x, y \in N$ справедливі умови

$$x \text{ exp } 0 = s(0), \tag{12.1}$$
$$x \text{ exp } s(y) = (x \text{ exp } y) \cdot x. \tag{12.2}$$

Тут знак $\cdot$ означає операцію множення на цій моделі. Для операції піднесення до степеня не існує теореми, подібної теоремі 3.5. Справді, взявши модель $M'' = (N''; a_0, t; =)$ з прикладу 3.3.3, дістаємо таку суперечність. Припустимо, що exp є операцією на M'', де $N'' = \{a_0, a_1\}$ і $t(a_0) = a_1$ та $t(a_1) = a_0$, яка задовольняє (12.1) і (12.2) для всіх $x, y \in N''$. Тоді із (12.2) випливає, що $a_0 \text{ exp } a_0 = (a_0 \text{ exp } a_1) \cdot a_0$, і тому $a_0 \text{ exp } a_0 = a_0$ на підставі (11.1). З іншого боку, із (12.1) випливає $a_0 \text{ exp } a_0 = t(a_0) = a_1$. Дістаємо суперечність із припущенням, яка показує, що індуктивна модель $M'' = (N''; a_0, t; =)$ не має жодної операції піднесення до степеня.

Але, застосовуючи теорему 3.4, можна показати, що кожна модель Пеано має єдину *операцію піднесення до степеня*. Дійсно, можна довести загальніший результат, звідки це твердження випливає як окремий випадок.

Теорема 3.6. *Нехай (N;0,s;=) – довільна модель Пеано, f – унарна операція на N і g – тернарна операція на N. Тоді існує одна і тільки одна бінарна операція j на N, яка для всіх $x, y \in N$ задовольняє умови*

$$j_x(0) = f(x), \tag{13.1}$$
$$j_x(s(y)) = g(x, y, j(x, y)). \tag{13.2}$$

Зокрема, коли j є така унарна операція на N, що $f(x) = s(0)$ для всіх $x \in N$, і g – така тернарна операція на N, що $g(x, y, z) = z \cdot x$ для всіх $x, y, z \in$ $\in N$, то дістаємо операцію піднесення до степеня на N. Дійсно, умови (13.1) і (13.2) дають $j(x, 0) = s(0)$ (тобто $x^0 = 1$) і $j(x, s(y)) = g(x, y, j(x, y)) =$ $= j(x, y) \cdot x$ (тобто $x^{y+1} = x^y \cdot x$).

Доведення теореми 3.6. Як випливає з доведення теореми 3.4, досить показати, що для кожного $x \in N$ існує єдина унарна операція j_x на N така, що для всіх $y \in N$

$$j_x(0) = f(x), \tag{14.1}$$
$$j_x(s(y)) = g(x, y, j_x(y)). \tag{14.2}$$

Показавши це, стає можливим доведення теореми 3.6 за допомогою загальних теоретико-множинних міркувань, які справедливі для всіх моделей (тобто без використання жодної з аксіом P1–P3).

Розглянемо множину $N_1 = N \times N$ усіх упорядкованих пар (y, z), де $y, z \in$ N. Нехай 0_x (де x довільний) елемент $(0, f(x)) \in N_1$, а s_x – унарна операція на N_1 така, що $s_x(y, z) = (s(y), g(x, y, z))$ для всіх $x, y \in N$. Застосовуючи теорему 3.4 до моделі Пеано $(N; 0, s; =)$ і моделі $(N_1; 0_x, s_x; =)$, отримуємо, що для кожного $x \in N$ існує єдине відображення h_x множини N у множину N_1 таке, що рівності

$$h_x(0) = 0_x, \tag{15.1}$$
$$h_x(s(y)) = s_x(h_x(y)) \tag{15.2}$$

мають місце для всіх $y \in N$.

Далі, нехай L і R такі відображення N_1 в N, що для всіх $x, y \in N$ $L(x, y) = x$, $R(x, y) = y$ і j_x та k_x (де $x \in N$ – довільний) – такі дві унарні операції, що $j_x(y) = R(h_x(y))$ і $k_x(y) = L(h_x(y))$ для всіх $y \in N$. Покажемо, що для всіх $y \in N$ операція j_x задовольняє умови (14.1) і (14.2). Для цього попередньо доведемо справедливість рівності $k_x(y) = y$ для всіх $y \in N$. Припустимо, що G – підмножина множини N, яка складається з тих елементів y, для яких $k_x(y) = y$. Оскільки $k_x(0) = L(h_x(0)) = L(0_x) = L(0, f(x)) = 0$, то $0 \in G$. Далі, нехай y – довільний елемент із G. Тоді на підставі (15.2), $k_x(s(y)) = L(h_x(s(y))) = L(s_x(h_x(y)))$. За означенням s_x маємо

$$\begin{aligned} s_x(h_x(y)) = (s(L(h_x(y))), g(x, L(h_x(y)), R(H_x(y)))) = \\ = (s(k_x(y)), g(x, k_x(y), j_x(y))) \end{aligned}$$

так, що $k_x(s(y)) = L(s_x(h_x(y))) = s(k_x(y))$. Оскільки $y \in G$, то звідси $k_x(s(y)) = s(y)$ і тому $s(y) \in G$.

Отже, множина G замкнута відносно s, і оскільки аксіома P3 справедлива для моделі $(N; 0, s; =)$, то $G = N$, тобто $k_x(y) = y$ для всіх $y \in N$.

Повертаючись знову до отриманої формули

$$s_x(h_x(y)) = (s(k_x(y)), g(x, k_x(y), j_x(y))),$$

бачимо, що тепер її можна спростити таким способом:

$$s_x(h_x(y)) = (s(y), g(x, y, j_x(y))).$$

Оскільки $j_x(s(y)) = R(h_x(s(y))) = R(s_x(h_x(y)))$, то відразу отримуємо, що $j_x(s(y)) = g(x, y, j_x(y))$, а це показує, що j_x задовольняє умову (14.2).

З іншого боку, $j_x(0) = R(h_x(0)) = R(0_x) = f(x)$, так що j_x задовольняє умову (14.1).

Для закінчення доведення теореми 3.6 залишається показати, що j_x єдина унарна операція на N, яка задовольняє цим двом умовам. Доведення виконують за тією самою схемою, що і доведення єдиності операції додавання в теоремі 3.4.

3.4.1. Моделі Пеано й індуктивні моделі

Чому операції додавання і множення існують в кожній індуктивній моделі, а операцію піднесення до степеня можна гарантувати лише в моделях Пеано? Щоб відповісти на це питання потрібно зрозуміти, як пов'язані між собою моделі Пеано та загальніші індуктивні моделі. На допомогу приходить алгебра.

Нехай $\mathcal{M} = (N; 0, s; =)$ і $\mathcal{M}_1 = (N_1; 0_1, s_1; =)$ – довільні дві моделі. Говорять, що $\mathcal{M}_1$ є гомоморфним образом $\mathcal{M}$, якщо існує гомоморфізм h моделі $\mathcal{M}$ на модель $\mathcal{M}_1$, тобто функція h, областю визначення якої є множина $\mathcal{M}$ і областю значень – вся множина $\mathcal{M}_1$, така, що $h(0) = 0$ і $h(s(x)) = s_1(h(x))$ для всіх $x \in M$.

Теорема 3.7. *Нехай $\mathcal{M} = (N; 0, s; =)$ – модель Пеано і $\mathcal{M}_1 = (N_1; 0_1, s_1; =)$ – довільна інша модель. Щоб модель $\mathcal{M}_1$ була гомоморфним образом $\mathcal{M}$, необхідно і достатньо, щоб вона була індуктивною моделлю.*

Доведення необхідності. Припустимо, що $\mathcal{M}_1$ – гомоморфний образ $\mathcal{M}$ і нехай h – гомоморфізм $\mathcal{M}$ на $\mathcal{M}_1$. Нехай G_1 – довільна підмножина N_1 така, що $0 \in G_1$ і G_1 замкнута відносно s_1. Щоб довести, що $\mathcal{M}_1$ є індуктивною моделлю, достатньо показати, що $G_1 = N_1$.

Із цією метою розглянемо підмножину G з N, яка включає ті і тільки ті елементи x, для яких $h(x) \in G_1$. Оскільки $h(0) = 0$, то $0 \in G$. Множина G замкнута відносно s. Дійсно, нехай $x \in G$ такий, що $h(x) \in G_1$. Тоді

$s_1(h(x)) \in G_1$ на підставі замкнутості G_1 відносно s_1. Але $h(s(x)) = s_1(h(x))$, оскільки відображення h гомоморфізм. Таким чином, $h(s(x)) \in G_1$ і, отже, $s(x) \in G$, а це означає замкнутість G відносно s. На підставі того, що в моделі $(N; 0, s; =)$ виконується аксіома P3, то $G = N$. А це означає, що $h(x) \in G_1$ для всіх $x \in N$. З того, що область визначення h збігається з N, а область значень – N_1, дістаємо, що $G_1 = N$.

Доведення достатності. Припустимо, що $(N_1; 0_1, s_1; =)$ – індуктивна модель. Оскільки $(N; 0, s; =)$ – модель Пеано, то можна застосувати теорему 3.4, з якої випливає, що існує (єдиний) гомоморфізм h із $\mathcal{M}$ в $\mathcal{M}_1$. Щоб закінчити доведення, залишається показати, що областю значень функції h є вся множина N_1.

Зрозуміло, що область значень функції h включає 0_1 на підставі того, що $h(0) = 0_1$. Вона замкнута відносно s_1, оскільки для довільного елемента z з області значень h повинен існувати елемент $x \in N$ такий, що $h(x) = z$. А звідси випливає, що $s_1(z) = s_1(h(x)) = h(s(x))$ і тому $s_1(z)$ теж належить області значень h. Але $\mathcal{M}_1$ задовольняє аксіому P3 і тому область значень h є N_1, що і потрібно було довести. ■

З теореми 3.7 випливає важливий і добре відомий наслідок.

Теорема 3.8. *Довільні дві моделі Пеано ізоморфні.*

Доведення. Нехай $\mathcal{M}$ і $\mathcal{M}_1$ – довільні дві моделі Пеано. За теоремою 3.7 існує гомоморфізм h із $\mathcal{M}$ в $\mathcal{M}_1$ і гомоморфізм h_1 із $\mathcal{M}_1$ в $\mathcal{M}$. Але суперпозиція двох гомоморфізмів теж є гомоморфізмом і тоді функція $h_1 * h$, тобто унарна операція на N, визначена для всіх $x \in N$ рівністю $h_1 * h(x) = h_1(h(x))$, гомоморфізм $\mathcal{M}$ в $\mathcal{M}$. Але на підставі теореми 3.4 існує лише єдиний гомоморфізм $\mathcal{M}$ в $\mathcal{M}$ і таким гомоморфізмом є тотожна операція на $\mathcal{M}$. Отже, $h_1 * h(x) = x$ для всіх $x \in M$. Звідси випливає, що h – бієкція і тому h – ізоморфізм моделей $\mathcal{M}$ і $\mathcal{M}_1$.

Принципове значення цієї теореми полягає в такому математичному наслідку.

Наслідок 3.1. *Якщо взяти якусь досить широку множину тверджень, яка включає символи “N”, “0”, “s” і “=”, то довільне твердження із цієї множини, справедливе в одній моделі Пеано, буде справедливим і для довільної іншої моделі Пеано.*

Відтак, якщо твердження із цієї множини справедливе для деякої окремої моделі Пеано, наприклад для системи $(N^*; 0^*, s^*; =)$ натуральних чисел, то воно буде справедливе і для всіх інших моделей Пеано і, отже, буде логічним наслідком аксіом P1–P3. Множина тверджень, до якої застосовний цей висновок, включає всі ті твердження про моделі, про які йшлося вище.

Таким чином, в подальшому замість того, щоб говорити про довільну модель Пеано, будемо говорити про систему $(N^*; 0^*, s^*; =)$ натуральних чисел.

Теорема 3.5 показала, що для довільної індуктивної моделі $\mathcal{M} = (N; 0, s; =)$ існує єдина операція додавання "$+$". А теорема 3.7 вказала на тісний зв'язок моделі $\mathcal{M}$ із системою натуральних чисел $\mathcal{N}^* = (N^*; 0^*, s^*; =)$. Розглянемо відношення між операцією "$+$" в $\mathcal{M}$ і операцією додавання "$+$" в системі $\mathcal{N}^* = (N^*; 0^*, s^*; =)$. Цей розгляд приводить до такої теореми.

Теорема 3.9. *Нехай $\mathcal{M} = (N; 0, s; =)$ – довільна індуктивна модель і "$+$" – її операція додавання. Нехай h – однозначно визначений гомоморфізм, який відображає $\mathcal{N}^*$ в $\mathcal{M}$. Тоді для всіх $x, y, \in \mathcal{N}^*$ виконується властивість $h(x +^* y) = h(x) + h(y)$.*

Доведення. Нехай x – довільний елемент множини $\mathcal{N}^*$ і G така підмножина $\mathcal{N}^*$, що $y \in G$ тоді і тільки тоді, коли $h(x +^* y) = h(x) + h(y)$. Оскільки $h(x +^* 0^*) = h(x) = h(x) + 0 = h(x) + h(0^*)$, то $0^* \in G$. Далі, нехай y – довільний елемент із G, тоді $h(x +^* y) = h(x) + h(y)$ і

$$h(x +^* s^*(y)) = h(s^*(x +^* y)) = s(h(x +^* y)) = s(h(x) + h(y)) = \\ = h(x) + s(h(y)) = h(x) + h(s^*(y))$$

і тому $s^*(y) \in G$. А із замкнутості множини G відносно s^* на підставі аксіоми P3 (справедливої в $\mathcal{N}^*$) дістаємо $G = N^*$. ■

Відповідна теорема для множення також справедлива і може бути доведена тим самим способом. Добре відомий такий загальний результат:

якщо деяке рівняння тотожно виконується в одній системі, то воно буде виконуватися і в довільному її гомоморфному образі [10].

Звідси випливає, що такі закони, як асоціативність, комутативність і дистрибутивність, справедливі для операцій додавання і множення в $\mathcal{N}^*$, будуть справедливі для операцій "$+$" і "$\cdot$" в довільній індуктивній моделі.

3.4.2. Гомоморфізми і конгруентності

З алгебри добре відомо, що з кожним гомоморфізмом моделей пов'язане відношення конгруентності.

Нехай h – довільний гомоморфізм моделі $\mathcal{M} = (N; 0, s; =)$ в деяку іншу модель. Тоді відношення R_h таке, що для довільних $x, y \in N$ xR_hy тоді і тільки тоді, коли $h(x) = h(y)$, є конгруентністю. Навпаки, для кожного відношення конгруентності R у моделі $\mathcal{M}$ існує гомоморфізм h, який відображає $\mathcal{M}$ на деяку іншу модель $\mathcal{N}_R$ так, що $R_h = R$.

Тепер можна визначити **фактор-модель** $\mathcal{N}_R = (N/R; [0]_R, s_R; =)$, де $s_R([x]_R) = [s(x)]_R$ для всіх $x \in N$ і $x \in [x]_R$, тут $[x]_R$ – клас еквівалентності елемента x.

Якщо визначити h як функцію, яка відображає N на N/R і таку, що $h(x) = [x]_R$ для всіх $x \in N$, то легко збагнути, що h – гомоморфізм $\mathcal{N}$ на $\mathcal{N}_R$ і що $R_h = R$, тобто з xRy випливає $h(x) = h(y)$ і навпаки.

Якщо h_1 і h_2 – гомоморфізми $\mathcal{M}$ на моделі $\mathcal{N}_1$ і $\mathcal{N}_2$ і коли $R_1 = R_2$, то моделі $\mathcal{N}_1$ і $\mathcal{N}_2$ ізоморфні. Звідси випливає, що кожний гомоморфний образ моделі $\mathcal{M}$ ізоморфний одній із моделей $\mathcal{N}_R$, визначеній деяким відношенням конгруентності R на N.

На підставі теорем 3.7 і 3.8 кожна індуктивна модель ізоморфна моделі $\mathcal{N}^*_R$, визначеній деяким відношенням конгруентності R в системі $\mathcal{N}^*$ натуральних чисел.

Наведемо опис усіх відношень конгруентності на моделі $\mathcal{N}^*$ в термінах відношення порядку "$<$" на $\mathcal{N}^*$.

Нехай m, n – довільні елементи з множини N^* натуральних чисел. Визначимо відношення конгруентності $R_{m,n}$ на N^* такими правилами: $xR_{m,n}y$ тоді і тільки тоді, коли справедлива одна з двох умов:

1) $x, y < n$ і $x = y$;

2) $x, y \geq n$ і для деякого $z \in N^*$ виконується або $x = y +^* (z \cdot^* m)$, або $y = x +^* (z \cdot^* m)$.

Теорема 3.10. *Бінарне відношення R на множині N^* буде відношенням конгруентності в моделі $\mathcal{N}^*$ тоді і тільки тоді, коли воно або відношення тотожності в $\mathcal{N}^*$, або існують числа $m, n \in N^*$ такі, що $R = R_{m,n}$.*

Доведення пропонується як вправа для самостійного виконання.

Відношення конгруентності $R_{m,n}$ є добре відомим відношенням порівнюваності за модулем. Індуктивна модель $\mathcal{N}^*_{R_{m,n}}$, яка відповідає такому відношенню, є просто системою класів лишків за модулем m, а операції додавання і множення визначаються на класах еквівалентності.

Очевидно, що коли R – одне з модулярних відношень конгруентності $R_{m,n}$, то s^*_R є просто деякою перестановкою елементів N^*. Отже, $\mathcal{N}_R$ в цьому випадку задовольняє аксіому P2. Крім того, якщо R – одне з відношень конгруентності $R_{m,n}$ для $n > 0$, то зрозуміло, що $[0^*]_R \neq s^*_R([x]_R)$ для всіх $x \in N^*$, і в цьому випадку $\mathcal{N}^*_R$ задовольняє аксіому P1. Таким чином, кожна індуктивна модель задовольняє аксіоми P1 і P2, про що йшлося вище.

Знайдемо необхідні і достатні умови існування гомоморфних образів операцій при гомоморфізмі h.

Нехай f – довільна бінарна операція на N^*. Якщо R є довільним відношенням конгруентності на $\mathcal{N}^*$, то, взагалі кажучи, не існує жодної бінарної операції g на N^*/R, яка є гомоморфним образом операції f при гомоморфізмі h, який відповідає відношенню R. Зі сказаного вище випливає, що необхідною і достатньою умовою існування гомоморфної операції g служить те, що для всіх $x, x_1, y, y_1 \in N^*$ таких, що xRx_1 і yRy_1, справедлива умова $f(x, y)Rf(x_1, y_1)$. Якщо ця умова виконується, то g буде такою операцією на N^*, що $g([x]_R, [y]_R) = [f(x, y)]_R$ для всіх $x, y \in N^*$.

Наприклад, хоча $2 \equiv 2 \pmod 3$ і $0 \equiv 3 \pmod 3$, але $2^0 \not\equiv 2^3 \pmod 3$. Отже, операція піднесення до степеня в N^* не має гомоморфного образу в $\mathcal{N}^*_{R_{3,0}}$. А операції $+^*$ і $\cdot^*$ є прикладами так званих *універсальних операцій* у тому сенсі, що вони є операціями f, які мають таку властивість, що для довільного відношення конгруентності R на N^* $f(x, y)Rf(x_1, y_1)$, якщо x, y, x_1, y_1 такі елементи з N^*, що xRx_1 і yRy_1. Саме на підставі цієї обставини кожна індуктивна модель має операції додавання і множення.

Нехай тепер операція j на N^* отримана застосуванням примітивної рекурсії з універсальних операцій f і g. Чи буде ця операція універсальною? Приклад операції піднесення до степеня показує, що це може і не бути універсальною операцією. Але коли j виявиться комутативною операцією, то вона буде універсальною (це можна показати узагальненням доведення теореми 3.5). Отже, наслідком некомутативності операції піднесення до степеня на множині N^* натуральних чисел є неможливість вивести існування операції піднесення до степеня в кожній індуктивній моделі за допомогою розповсюдження доведення теореми 3.5.

Розглянута властивість комутативності є достатньою умовою універсальності, але не є необхідною умовою. Наприклад, операція j така, що $j(x, y) = x^2 \cdot y$ для всіх $x, y \in N^*$, є універсальною операцією, яка отримана застосуванням примітивної рекурсії з універсальних функцій f і g таких, що $f(x) = 0$ і $g(x, y, z) = z + x^2$ для всіх $x, y, z \in N^*$, але j не комутативна.

Підсумовуючи, зауважимо, що основою застосування методу визначення за індукцією в моделі $\mathcal{M}$ є існування єдиного гомоморфізму $\mathcal{M}$ в яку-небудь іншу модель. Отже, теорема 3.4 служить основою всіх визначень за математичною індукцією для моделей Пеано.

Виявляється, що ця властивість характерна для моделей Пеано – *єдиних моделей, в яких всі визначення за математичною індукцією можуть бути обґрунтовані.*

3.5. Індуктивні моделі в ПЗ

З огляду на доведені факти для індуктивних моделей і моделей Пеано, наведемо приклади інтеграції цих моделей у ПЗ.

Нагадаємо, що означення за індукцією має такий узагальнений вигляд:

1. *Базис:* Задано список специфічних (базисних) елементів множини A.

2. *Індукція:* Задано одне або декілька правил побудови нових елементів множини A з існуючих елементів в A.

3. *Замкнутість:* Множина A складається з тих і тільки тих елементів, які отримані з базисних і побудовані за індуктивними правилами.

Замкнутість є важливою частиною означення за індукцією. Якщо цей крок відсутній, то може існувати велика кількість множин, які задовольняють перші два кроки індуктивного означення. Наприклад, якщо множини $\mathcal{N}$ і $\{3, 5, 7, \ldots\}$ обидві включають елемент 3 і x їхній спільний елемент, то яким буде елемент $x + 2$.

3.5.1. Абстрактний тип даних "Натуральне число"

Індуктивне означення множини натуральних чисел, яку позначатимемо $\mathcal{N}$, було наведено вище. Повторимо це означення ще раз для наглядності.

Індуктивне означення множини натуральних чисел $\mathcal{N}$ набуває вигляду:

Базис: $0 \in \mathcal{N}$.

Індукція: якщо $n \in \mathcal{N}$, то $n + 1 \in \mathcal{N}$.

конструкторами множини $\mathcal{N}$ є число 0 і операція додавання 1 до елемента з $\mathcal{N}$. Операція додавання 1 до n називалася "слідує за" і позначалася s. Використовуючи цю операцію, крок індукції можна записати таким чином:

якщо $n \in \mathcal{N}$, то $s(n) \in \mathcal{N}$,

і тепер можна сказати, що $\mathcal{N}$ індуктивна множина з двома конструкторами 0 і s. Це підтверджує також теорема 2.6: довільна підмножина B множини $\mathcal{N}$, побудована за вказаними правилами, збігається з $\mathcal{N}$.

Надалі будемо називати **множину індуктивною**, якщо вона має індуктивне означення.

Приклад 3.5.1. Визначити індуктивно множину $A = \{1, 3, 7, \ldots\}$. Тут конструкторами виступають число $1 \in \mathcal{N}$ і якщо $x \in A$, то $2x + 1 \in A$. Отже, індуктивне означення буде таким:

Базис: $1 \in A$.
Індукція: якщо $x \in A$, то $2x + 1 \in A$.

На підставі теореми 2.6 дістаємо, що множина A індуктивна. ♠

Приклад 3.5.2. Чи є множина $A = \{2, 3, 4, 7, 8, 11, 15, 16, \ldots\}$ індуктивною?

З першого погляду важко визначити індуктивність чи неіндуктивість множини A. Але коли її подати у вигляді об'єднання двох множин B і C, де

$$B = \{2, 4, 8, 16, \ldots\} \text{ і } C = \{3, 7, 11, 15, \ldots\},$$

то обидві ці множини очевидно є індуктивними: конструкторами множини B виступають число 2 і операція множення на $2 - 2x$, а для множини C конструкторами будуть число 3 і операція додавання числа $4 - x + 4$. Оскільки операції додавання і множення в індуктивних множинах єдині, то можна скомбінувати ці два означення в одне індуктивне означення множини A:

Базис: $2, 3 \in A$.

Індукція: якщо $x \in A$ і x непарне, то $x + 4 \in A$, в іншому разі $2x \in A$. ♠

Приклад 3.5.2 показує, що може існувати не один базисний елемент і крок індукції може включати перевірку істинності умови. Але виникає питання: *коли об'єднання двох індуктивних множин, буде індуктивною множиною?* З прикладу видно, що множини базисних елементів не перетинаються й індуктивні умови є взаємовиключальними, тобто такими, що на тому самому елементі ці умови не можуть бути істинними одночасно. Саме ці властивості дають можливість сформулювати індуктивне означення об'єднання індуктивних множин, яке гарантує єдиність результуючої множини.

Крім того, це означення гарантує і обчислювальність елементів такого типу множин. Дійсно, обчислення елементів індуктивної множини випливає з поданого нижче факту теорії примітивно-рекурсивних функцій.

Нехай задано деякі функції $f_i(x_1, \ldots, x_n),\ i = 1, \ldots, k+1$, і вказано умови (предикати) $u_j(x_1, \ldots, x_n),\ j = 1, 2, \ldots, k$, які можуть бути для довільної n-ки чисел $x_1, \ldots, x_n$ істинними або хибними.

Припустимо, що для жодної n-ки чисел ніякі дві умови не можуть бути хибними одночасно (такі умови називаються **взаємовиключальними)**. Функцію $f(x_1, \ldots, x_n)$, задану схемою

$$f(x_1, \ldots, x_n) = \begin{cases} f_1(x_1, \ldots, x_n), & \text{коли} \quad u_1(x_1, \ldots, x_n) \ = \ 0; \\ \ldots\ldots\ldots\ldots & \ldots\ldots \quad \ldots\ldots\ldots\ldots \quad \ldots \ \ldots \\ f_k(x_1, \ldots, x_n), & \text{коли} \quad u_k(x_1, \ldots, x_n) \ = \ 0; \\ f_{k+1}(x_1, \ldots, x_n) & \text{для решти} \quad x_1, \ldots, x_n, \end{cases}$$

називають **кусково заданою**.

Серед функцій натурального аргументу виділяють примітивно рекурсивні функції, з означенням яких можна познайомитися у [6]. Для кусково заданих примітивно рекурсивних функцій справедлива

Теорема 3.11. *Нехай задані n-арні примітивно-рекурсивні функції $f_1, \ldots, f_{k+1}, u_1, \ldots, u_k$, де u_i – взаємовиключальні умови, $i = 1, 2, \ldots, k$. Тоді кусково задана функція*

$$f(x_1, \ldots, x_n) = \begin{cases} f_1(x_1, \ldots, x_n), & \textit{коли} \quad u_1(x_1, \ldots, x_n) \ = \ 0; \\ \ldots\ldots\ldots\ldots & \ldots\ldots \quad \ldots\ldots\ldots\ldots \quad \ldots \ \ldots \\ f_k(x_1, \ldots, x_n), & \textit{коли} \quad u_k(x_1, \ldots, x_n) \ = \ 0; \\ f_{k+1}(x_1, \ldots, x_n) & \textit{для решти} \quad x_1, \ldots, x_n. \end{cases}$$

теж примітивно-рекурсивна функція.

Доведення. Дану функцію можна подати у вигляді

$$f = f_1 \cdot (1 - u_1) + \ldots + f_k \cdot (1 - u_k) + f_{k+1} \cdot (u_1 \cdot u_2 \cdot \ldots \cdot u_k).$$

З примітивної рекурсивності функцій обмеженої різниці, додавання і множення [6], випливає примітивна рекурсивність функції f. ■

Зауважимо, що умови u_i в теоремі 3.11 мають вигляд $u_i = 0$. Оскільки умови $p_i = p_j,\ p_i \le p_j,\ p_i < p_j$ еквівалентні умовам

$$|p_i - p_j| = 0,\ p_i \div p_j = 0,\ 1 \div (p_j \div p_i) = 0,$$

то теорема 3.11 залишається справедливою і в цих випадках, якщо $p_i,\ p_j$ – примітивно рекурсивні функції.

Отже, коли індуктивні правила включають лише примітивно-рекурсивні функції, то обчислення предиката належності елемента до множини алгоритмічно розв'язується.

3.5.2. Абстрактний тип даних “Список (list)”

Одним із важливих абстрактних типів даних у програмуванні є списки. Нехай A – деяка множина а $\langle\rangle$ означає порожний список. Тоді індуктивне означення АТД список $List(A)$ набуває вигляду:

Базис: $\langle\rangle \in List(A)$.
Індукція: якщо $x \in A$ і $t \in List(A)$, то $cons(x, t) \in List(A)$,
де $cons$ означає операцію конкатенації (добутку) списків. Отже, конструкторами виступають порожній список $\langle\rangle$ і операція конкатенації $cons$.

Приклад 3.5.3. Нехай $A = \{a, b\}$. Скористаємося індуктивним означенням, щоб побачити як деякі списки стають елементами множини $List(A)$.

Базис означення говорить про те, що $\langle\rangle \in List(A)$. Оскільки $a \in A$ і $\langle\rangle \in$ $\in List(A)$, то крок індукції будує список $\langle a\rangle = cons(a, \langle\rangle) \in List(A)$. Аналогічно $\langle b\rangle = cons(b, \langle\rangle) \in List(A)$. Далі, оскільки $a \in A$ і $\langle a\rangle \in List(A)$, то крок індукції дає список $\langle a, a\rangle \in List(A)$. Таким способом до множини $List(A)$ додаються списки $\langle b, a\rangle$, $\langle a, b\rangle$, $\langle b, b\rangle$, $\langle a, b, a\rangle, \ldots$ ♠

Велика кількість задач у мовах програмування пов'язана з обробленням даних, які представлені списками. Для маніпуляції з такими даними на списках визначаються інші операції та функції, деякі з яких іноді називають *деструкторами*, оскільки вони потребують окремих частин списку. Зокрема, до них належать такі:

- $head(p)$ – голова списку,
- $tail(p)$ – хвіст списку,
- $div(p, i)$ – розбити список на два підсписки по i-му елементу,

• $del(p, i)$ – видалити i-й елемент зі списку,

• $add(p, i, x)$ – вставити елемент між i-м і $(i + 1)$-м елементами списку,

• $rev(p)$ – записати елементи списку у зворотному порядку,

• $l(p)$ – довжина списку,

• $sublist(p, i, j)$ – підсписок, який починається з i-го елементу списку довжини $j - i$ та інші.

Зазначимо, що операції $head, tail$ і $cons$ утворюють базис засобів для маніпуляції зі списками. Деталі індуктивних означень наведених операцій і функцій на індуктивній множині “Список” представлені на стор. 81.

Приклад 3.5.4. Нехай потрібно побудувати множину S всіх непорожніх списків над алфавітом $\{0, 1\}$, де елементи 0, 1 у списках альтернуються між собою, тобто

$$S = \{\langle 0\rangle, \langle 1\rangle, \langle 1, 0\rangle, \langle 0, 1\rangle, \langle 0, 1, 0\rangle, \langle 1, 0, 1\rangle, \ldots\}.$$

Якщо покласти 0 і 1 базисними елементами, то можна конструювати нові списки із списків $x \in S$ за допомогою перевірки істинності умови: $head(x)$ дорівнює 0, чи дорівнює 1. Якщо $head(x) = 0$, то додаємо зліва до списку елемент 1, інакше додаємо до списку елемент 0. Тоді індуктивне означення множини S набуває вигляду

Базис: $\langle 0\rangle, \langle 1\rangle \in S$.

Індукція: якщо $x \in S$ і $head(x) = 0$, то $cons(1, x) \in S$, в іншому разі $cons(0, x) \in S$.

Приклад 3.5.5. Визначити множину списків над множиною $A = \{a, b\}$, які починаються елементом a, а потім іде 0 або більше входжень у список елемента b, тобто

$$S = \{\langle a\rangle, \langle a, b\rangle, \langle a, b, b\rangle, \langle a, b, b, b\rangle, \ldots\}.$$

Зрозуміло, що базисним елементом потрібно взяти список $\langle a\rangle$. Далі конструювання нового списку з деякого списку $x \in S$ відбувається так: до хвоста списку x зліва додається елемент b, а потім зліва до отриманого списку додають елемент a. Отже, індуктивне означення множини S набуває вигляду

Базис: $\langle a\rangle \in S$.

Індукція: якщо $x \in S$, то $cons(a, cons(b, tail(x)) \in S$.

Це означення використовує властивість операції $tail(x) = \langle\rangle$, якщо x базисний елемент, або одноелементний список.

Приклад 3.5.6. (*Узагальнені списки*) Узагальненим списком називають список над алфавітом A, який може включати списки як елементи. Наприклад, нехай $A = \{a, b\}$, тоді $GenList(A)$ включає такі елементи:

$$\langle\langle b, a\rangle, b\rangle, \langle\langle\langle b\rangle\rangle, a\rangle, \langle a, b, a\rangle, \ldots.$$

Індуктивне означення множини $GenList(A)$, очевидно, потребує узагальнення операції $cons$ так, щоб її аргументом були списки, а не тільки елементи множини A. Якщо визначити операцію $cons$ так, щоб її першим аргументом був список, то можна будувати списки $\langle\langle\rangle\rangle$ із $\langle\rangle$ таким способом $cons(\langle\rangle, \langle\rangle)$. Подібно можна отримати списки $\langle\langle a\rangle\rangle$ з $\langle a\rangle$ і $\langle\rangle$, тобто $cons(\langle a\rangle, \langle\rangle)$, а також побудувати списки:
$cons(a, \langle b\rangle) = \langle a, b\rangle$, $\quad cons(\langle a\rangle, \langle b\rangle) = \langle\langle a\rangle, b\rangle$,
$cons(a, \langle\langle b\rangle\rangle) = \langle a, \langle b\rangle\rangle$, $\quad cons(\langle a\rangle, \langle\langle b\rangle\rangle) = \langle\langle a\rangle, \langle b\rangle\rangle$.

Якщо A – деякий алфавіт, то індуктивне означення множини узагальнених списків $GenList(A)$ набуває вигляду

Базис: $\langle\rangle \in GenList(A)$.

Індукція: якщо $x \in A \cup GenList(A)$ і $t \in GenList(A)$, то $cons(x, t) \in \in GenList(A)$.

Узагальнені версії операцій $head$ і $tail$ працюють так само, як для множини $List(A)$. Наприклад,

$$head(\langle\langle a, b\rangle, c\rangle) = \langle a, b\rangle,\ tail(\langle\langle a, b\rangle, c\rangle) = \langle c\rangle.\spadesuit$$

3.5.3. Абстрактний тип даних "Рядок (string)"

Означення даних типу "рядок (string)" в алфавіті A подібне до означення даних типу "список". Як базисний елемент виступає порожній рядок, який будемо позначати e, а конструктором – операція $append$, результатом виконання якої є дописування зліва до рядка елемента з A. Операцію $append$ позначають, зазвичай, точкою "$\cdot$". Наприклад, дописування літери a до рядка s має вигляд $a \cdot s$.

Якщо $s = aba$, то значення виразу $a \cdot s$ набуває вигляду

$$a \cdot s = aaba.$$

Коли символ дописується до порожнього рядка, то результатом буде цей символ, тобто

$$a \cdot e = e \cdot a = a.$$

Зазначимо, що операція $append$ задовольняє закон правої асоціативності, тобто $a \cdot b \cdot c = a \cdot (b \cdot c)$.

Тепер можна дати індуктивне означення типу даних "рядок".

Нехай A – деякий алфавіт, тоді індуктивне означення множини A^* всіх рядків над алфавітом A набуває вигляду

Базис: $e \in A^*$.

Індукція: якщо $a \in A$ і $s \in A^*$, то $a \cdot s \in A^*$.

Наприклад, якщо $A = \{a, b\}$, то рядок bab будується так:

$$\begin{aligned} bab \cdot e &= b \cdot a \cdot be \\ &= b \cdot a \cdot b \\ &= b \cdot ab \\ &= bab. \end{aligned}$$

На цьому типі даних визначаються операції $head$ і $tail$ так само, як вони визначалися для списків. Наприклад,

$$head(abc) = a \text{ і } tail(abc) = bc.$$

Приклад 3.5.7. Нехай $A = \{0, 1\}$ і потрібно визначити множину таких рядків L, що кожний рядок в L включає лише єдине входження елемента 0 справа в рядку. Наприклад,

$$L = \{0, 10, 110, 1110, 11110, \ldots\}.$$

Чи визначається множина L індуктивно? Так, визначається:

Базис: $0 \in L$.

Індукція: якщо $s \in L$, то $1 \cdot s \in L$.

Приклад 3.5.8. (*Дописування справа*) Нехай $A = \{0, 1\}$ і потрібно побудувати множину S рядків над алфавітом A з такою властивістю елементів: немає в S жодного рядка, голова якого 0, крім самого рядка 0. Тоді S включає такі рядки:

$$0, 1, 10, 11, 100, 101, 110, 111, \ldots.$$

Індуктивне означення множини S набуває вигляду

Базис: $0, 1 \in S$.

Індукція: якщо $s \in S$ і $s \neq 0$, то $s \cdot 0, s \cdot 1 \in S$.

Кілька перших рядків, побудованих за цим означенням, мають вигляд

$$0, 1, 10, 11, 100, 101, 110, 111, 1000, 1001, \ldots. \spadesuit$$

Приклад 3.5.9. Нехай $A = \{a, b\}$ – алфавіт і множина S над алфавітом A має такий вигляд:

$$S = \{a, b, ab, ba, aab, bba, aaab, bbba, \ldots\}.$$

Припустимо, що базисними рядками є a і b. З рядка a можна отримати рядки ba, а з рядка ba – рядок bba і т. д. Аналогічно з рядка b можна побудувати рядки ab і aab. Другий спосіб побудови множини S такий, як у прикладі 3.4.2: подати множину S у вигляді двох простіших множин S_1 і S_2, де

$$S_1 = \{a, ba, bba, \ldots\} \text{ і } S_2 = \{b, ab, aab, \ldots\}.$$

Щоб подати конструкцію множини S, скористаємося операцією $head$. Для заданого рядка $s \in S$, якщо $s = a$, то будуємо рядок ba і якщо $s = b$, то будуємо рядок ab.

В протилежному випадку, якщо $head(s) = a$, то будуємо $a \cdot s$, а якщо $head(s) = b$, то будуємо $b \cdot s$. Отже, індуктивне означення множини S можна дати у такий спосіб:

Базис: $a, b \in S$.
Індукція: якщо $s \in S$, то
якщо $s = a$, то $b \cdot a \in S$;
якщо $s = b$, то $a \cdot b \in S$;
якщо $s \neq a$ і $head(s) = a$, то $a \cdot s \in S$;
якщо $s \neq b$ і $head(s) = b$, то $a \cdot s \in S$. ♠

Вищенавеедені індуктивні означення і приклади ілюстрації цих означень описують три типи даних: натуральне число, список і рядок. Саме ці три типи даних становлять основні типи даних мови програмування Python [2]. Ці типи даних також присутні практично у всіх інших мовах програмування.

Другий висновок, який випливає зі сказаного, – це те, що всі моделі типів даних є індуктивними. Виняток являють собою моделі прикладів 3.4.1 і 3.4.2, які є моделями Пеано. Решта моделей є індуктивними, але не є моделями Пеано. Цей висновок випливає з того, що ці моделі не ізоморфні між собою. Наприклад, операція додавання в моделі Пеано задовольняє закон комутативності, а операція *cons* такий закон не задовольняє. Отже, ізоморфізму між цими моделями бути не може.

Якщо проаналізувати множини, які породжуються індуктивними означеннями, то вони рівнопотужні з множиною натуральних чисел N. Саме ця обставина служить основою застосування методу математичної індукції та індуктивних означень.

Умова індуктивності дозволяє застосовувати метод математичної індукції не тільки на добре упорядкованих множинах, а і на частково упорядкованих множинах і виконувати на них побудову за індукцією. Наступні АТД визначають саме таким способом.

3.5.4. Абстрактний тип даних "Бінарне дерево"

Користуючись вищенаведеними теоретичними підставами, розглянемо АТД "бінарне дерево".

Нагадаємо, що бінарним деревом називають ациклічний граф, кожна вершина якого має не більше двох синів.

Нехай $tree(L, x, R)$ означає бінарне дерево, у якого

- x – корінь,
- L – ліве піддерево,
- R – праве піддерево.

Зображатимемо бінарні дерева у вигляді упорядкованої трійки і будемо користуватися таким позначенням: $tree(L, x, R) = (L, x, R)$. Припустимо, що A – деяка множина. Нехай $BinTree(A)$ означає множину всіх бінарних дерев, вершини яких узяті з множини A. Тепер можна перейти до індуктивного означення множини $BinTree(A)$, використовуючи два конструктори $\langle\rangle$ (порожнє дерево) і $tree$:

Базис: $\langle\rangle \in BinTree(A)$.
Індукція: якщо $x \in A$ і $L, R \in BinTree(A)$, то $tree(L, x, R) \in BinTree(A)$.

Введемо також операції-деструктори для бінарних дерев. Припустимо, що

$$left, \ root \ і \ right$$

означають операції, результатом обчислення яких є відповідно ліве піддерево, корінь і праве піддерево непорожнього дерева. Наприклад, якщо $t = tree(L, x, R)$, то

$$left(t) = L, root(t) = x, right(t) = R.$$

Приклад 3.5.10. (*Спарене дерево*) Нехай потрібно працювати з множиною Twins бінарних дерев T, над множиною $A = \{0, 1\}$, які мають такі властивості: ліве і праве піддерева кожної вершини в T ідентичні за структурою і наповненням вершин. Наприклад, Twins включає порожнє дерево і довільне одновершинне дерево, а також дерева, які показані на рис. 3.4.1.

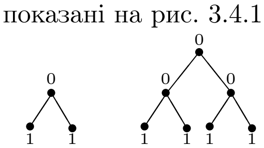


Рис. 3.5.1. Дерева Twins

Тепер можна дати індуктивне означення множини бінарних дерев Twins, в якій кожне нове дерево має однакові ліве і праве піддерева.

Базис: $\langle\rangle \in Twins$.
Індукція: якщо $x \in A$ і $T \in Twins$, то $tree(T, x, T) \in Twins$.

Приклад 3.5.11. (*Чергування*) Нехай $A = \{0, 1\}$ і Opps означає множину всіх непорожніх бінарних дерев T над A з такою властивістю: ліве і праве піддерева кожної вершини в T ідентичні, але їх корені, тобто вершини 0 і 1, чергуються. Наприклад, одновершинне дерево належить до Opps, як і нижченаведені дерева на рис. 3.5.2.

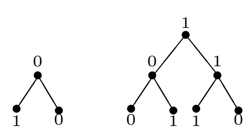


Рис. 3.5.2. Дерева Opps

Оскільки множина T не включає порожнього дерева, то два одно-вершинні дерева 0 і 1 становлять базис дерев у множині Opps. Тоді індуктивне означення множини Opps набуває вигляду

Базис: $tree(\langle\rangle, 0, \langle\rangle), tree(\langle\rangle, 1, \langle\rangle) \in Twins$.
Індукція: якщо $x \in A$ і $T \in Opps$, то
якщо $root(T) = 0$, то $tree(T, x, tree(right(T), 1, left(T))) \in Opps$,
в іншому разі $tree(T, x, tree(right(T), 0, left(T))) \in Opps$. ♠

3.5.5. Декартів добуток множин

Якщо йдеться про декартів добуток $\mathcal{N} \times \mathcal{N}$ множин натуральних чисел, то на підставі індуктивності $\mathcal{N}$ індуктивне означення декартового добутку можна подати у такому вигляді:

Перший спосіб
Базис: $(0, n) \in \mathcal{N} \times \mathcal{N}$ для всіх $n \in \mathcal{N}$.
Індукція: якщо $(m, n) \in \mathcal{N} \times \mathcal{N}$, то $(s(m), n) \in \mathcal{N} \times \mathcal{N}$.

Другий спосіб
Базис: $(0, 0) \in \mathcal{N} \times \mathcal{N}$.
Індукція: якщо $(m, n) \in \mathcal{N} \times \mathcal{N}$, то $(s(m), n), (m, s(n)) \in \mathcal{N} \times \mathcal{N}$.

Зауважимо, що це означення генерує деякі пари більше одного разу. Наприклад, (1,1) можна подати як (s(0),1) і як (1,s(0)). Отже, (1,1) буде наступником (0,1) і (1,0), а пари (0,1) і (1,0) будуть наступниками (0,0).

Розглянемо таке завдання: *дана індуктивно визначена множина A і деякий елемент d того самого типу, що і елементи множини A. Визначити належність/неналежність елемента d до множини A.* Для прикладу візьмемо індуктивну множину A, визначену в прикладі 3.5.2, і елемент $d = 127$. Визначити: належить, чи ні елемент 127 до множини C, тобто визначити істинність чи хибність предиката $d \in C$.

Для визначення істинності предиката $d \in C$ застосовуємо правила, за якими будувалася множина C. Елементи множини індуктивно породжувалися базисним елементом 3 й операцією $x + 4$. Отже, для перевірки істинності предиката $d \in C$ будемо породжувати елементи множини C доти, доки або не отримаємо елемент d, або не отримаємо елемент більший за елемент d. Звичайно, істинність предиката $127 \in C$ можна визначити пошуком розв'язку рівняння $127 = 3 + 4n$ у множині натуральних чисел. Оскільки $n = \frac{124}{4} = 31$ є розв'язком цього рівняння, то предикат $127 \in C$ істинний.

Звідси випливає такий важливий висновок: *проблема обчислення характеристичної функції (а це і є встановлення істинності/хибності предиката $x \in C$) розв'язується для множини C, яка визначена індуктивно.*

Але в загальному випадку питання визначення істинності предиката $d \in C$ залишається досить складним, особливо коли про множину C нічого невідомо, крім можливо типу її елементів.

ЗАДАЧІ ДЛЯ САМОСТІЙНОЇ РОБОТИ

1. Навести індуктивне означення наступних множин, маючи в розпорядженні тільки функцію $s : \mathcal{N} \to \mathcal{N}$ – "наступний":

а) множину непарних чисел; б) множину парних чисел;

в) множину $S = \{4, 7, 10, 13, \ldots\} \cup \{3, 6, 9, 12, \ldots\}$.

2. Дано непорожню множину A, навести індуктивне означення таких підмножин $List(A)$:

а) множину $Even$ всіх списків із парною кількістю елементів;

б) множину Odd всіх списків із непарною кількістю елементів.

3. Дано непорожню множину $A = \{a, b\}$, навести індуктивне означення множини S всіх списків над A, в яких альтернують між собою a і b, наприклад, $\langle\rangle, \langle a\rangle, \langle b\rangle, \langle a, b, a\rangle$ і $\langle b, a\rangle \in S$, а $\langle a, a\rangle \notin S$.

4. Навести індуктивне означення таких множин:

а) множину $Even$ всіх рядків в алфавіті $A = \{a, b\}$ парної довжини;

б) множину Odd всіх рядків в алфавіті $A = \{a, b\}$ непарної довжини;

в) множину Rat всіх рядків вигляду $a.b$, де a і b десяткові числа.

5. Дано множину рядків в алфавіті $A = \{a, b\}$, яка включає такі елементи:

$$S = \{a, b, ab, ba, aab, bba, aaab, bbba, \ldots\}.$$

Навести індуктивне означення множини S.

6. Користуючись наступними записами, навести графічні зображення бінарних дерев:

а) $tree(tree(\langle\rangle, x, \langle\rangle), y, tree(\langle\rangle, z, \langle\rangle, w, \langle\rangle))$;

б) використовуючи конструктори бінарних дерев, записати за їх допомогою бінарне дерево $\langle\langle\rangle, 3, \langle\langle\rangle, 4, \langle\rangle\rangle\rangle$.

7. Дано індуктивне означення підмножини $B \subset \mathcal{N} \times \mathcal{N}$:

Базис: $\langle 0, 0\rangle \in B$.

Індукція: якщо $\langle x, y\rangle \in B$, то $\langle s(x), y\rangle$ і $\langle s(x), s(y)\rangle \in B$.

а) Записати множину B у вигляді $B = \{\langle x, y \mid \text{властивість} P\rangle\}$;

б) Знайти елементи в множині B, які визначаються більше одного разу.

8. Довести, що індуктивно визначена множина буде зліченною, якщо множини базових і індуктивних правил скінченні.

3.6. Індуктивна побудова мов

Мова – це множина слів у деякому алфавіті. Якщо X – алфавіт, то множина всіх слів в алфавіті X позначається як X^*. Мовою в алфавіті X називають довільну підмножину множини X^*. Багато корисних мов визначають індуктивно. Наприклад, нехай розглядається множина десяткових чисел

$$Digits = \{0, 1, 2, 3, 4, 5, 6, 7, 8, 9\}.$$

Множину D десяткових чисел визначають індуктивно так:

Базис: $Digits \subset D$.

Індукція: якщо $d \in Digits$ і $n \in D$, то $d \cdot n \in D$,
де $\cdot$ означає операцію *append*.

Наведене індуктивне означення, наприклад, у мові програмування ПАСКАЛЬ визначає тип даних "ціле без знака".

Це означення не є однозначним, оскільки за ним можна дістати число 45, число 045, 0045 і т. д. Для забезпечення однозначності необхідно ввести деякі обмеження в означення. Ці обмеження досить очевидні:

Базис: $Digits \subset D$.

Індукція: якщо $n \in D$ і $d \in Digits$ і $n \neq 0$, то $n \cdot d \in D$.

Зазначимо, що множину X^* всіх слів в алфавіті X теж можна визначити індуктивно за допомогою операції $\cdot$:

Базис: $e \in X^*$.

Індукція: якщо $a \in X$ і $p \in X^*$, то $a \cdot p \in X^*$.

Приклад 3.6.1. (*Ідентифікатор*) Нехай $X = \{A, B, \ldots, Z, a, b, \ldots, z\} \cup$ $\cup \{0, 1, 2, 3, 4, 5, 6, 7, 8, 9\} = S \cup Digits$ – алфавіт. Індуктивне означення множини ідентифікаторів $Iden$ мови програмування набуває вигляду

Базис: $S \in Iden$.

Індукція: якщо $x \in S$ і $d \in D \cup Iden$, то $x \cdot d \in Iden$,
де D – множина десяткових чисел, визначена вище. ♠

Оскільки мови є множинами, то до мов можна застосовувати теоретико-множинні операції – об'єднання, перетину, різниці та доповнення. Другою важливою можливістю комбінування мов з уже побудованих, є операція добутку мов. Нехай L і M дві мови, тоді мова $L \cdot M$ має вигляд

$$L \cdot M = \{s \cdot t \mid s \in L \wedge t \in M\}.$$

Операція добутку мов дає можливість визначати мови через уже побудовані мови. Наприклад, можна визначити множину слів вигляду $a.b$, де a і b десяткові числа, використовуючи добуток трьох мов: $\{.\}, Digits, Digits^*$:

$$Digits \cdot Digits^* \cdot \{.\} \cdot Digits \cdot Digits^*.$$

Операція добутку мов має багато корисних властивостей. Основними серед них є такі: для довільних мов L, M, N

а) $L \cdot \{e\} = \{e\} \cdot L = L$; б) $L \cdot \emptyset = \emptyset \cdot L = \emptyset$;
в) $L \cdot (M \cup N) = L \cdot M \cup L \cdot N$; $(M \cup N) \cdot L = M \cdot L \cup N \cdot L$;
г) $L \cdot (M \cap N) = L \cdot M \cap L \cdot N$; $(M \cap N) \cdot L = M \cdot L \cap N \cdot L$.

Індуктивне означення степеня L^n мови L має вигляд

Базис: $\{e\} = L^0 \in L^n$, де $n \geq 1$.
Індукція: якщо $i < n$ і $L^i \in L^n$, то $L \cdot L^i \in L^n$.

Наприклад, якщо $L = \{a, bb\}$, то

$$L^0 = \{e\}, L^1 = \{a, bb\}, L^2 = L \cdot L = L^2 = \{aa, abb, bba, bbbb\}, \ldots.$$

Тепер ітерація L^* мови L набуває вигляду

$$L^* = L^0 \cup L^1 \cup L^2 \cup \ldots \cup L^n \cup \ldots.$$

Позитивну ітерацію L^+ мови L виражають через мови L^n у такий спосіб:

$$L^+ = L^1 \cup L^2 \cup L^3 \cup \ldots.$$

Отже, $L^* = L^+ \cup \{e\}$.

Граматики і мови. Неформально під граматикою G розуміють множину правил, за якими визначають структуру слів деякої мови. Граматики важливі в *computer science* не тільки для визначення мов програмування, а і для знаходження множин даних для програм. У визначенні мов програмування типовим завданням є побудова алгоритму, який визначає належність/неналежність програми (як слова в алфавіті мови програмування) до слів цієї мови. Отже, основною проблемою тут є проблема обчислення предиката $p \in L(G)$, де $L(G)$ – мова, породжена граматикою G.

Інколи легко написати граматику (тобто її правила), але інколи це досить важке завдання. Найбільш важливим аспектом написання граматики є знання про мову, яку потрібно побудувати. Якщо G – граматика, то мова $L(G)$ є множиною T^* термінальних слів, виведених із початкового символу σ граматики G. Отже, можна написати

$$L(G) = \{s \mid s \in T^* \wedge \sigma \to^* s\}.$$

Правило граматики будемо називати *рекурсивним*, якщо ліва частина правила входить в його праву частину. Наприклад, правило $\sigma \to a\sigma$ – рекурсивне. Правило $\sigma \to \alpha$ називають *непрямо рекурсивним*, якщо серед правил виведення α є правило, яке включає σ. Прикладом є граматика

$$\begin{aligned} \sigma &\to b \mid aA, \\ A &\to c \mid b\sigma. \end{aligned}$$

Правила $\sigma \to aA$ і $A \to a\sigma$ обидва непрямо рекурсивні, оскільки маємо такі виведення:

$$\begin{aligned} \sigma &\to aA \to ab\sigma, \\ A &\to b\sigma \to baA. \end{aligned}$$

Умова I: Далі припускається, що всі правила граматики мають єдиний нетермінал, що є лівою частиною правила.

Граматику називають *рекурсивною*, якщо серед її правил є або рекурсивні правила, або непрямо рекурсивні правила. Звідси випливає таке твердження:

граматика для нескінченної мови повинна бути рекурсивною.

Подивимося тепер з іншого боку на проблему створення мови за допомогою граматики. Відомо, що за означенням мова, породжена граматикою, є множиною слів, побудованих за правилами граматики. Тут теж можна помітити і стверджувати таке:

Теорема 3.12. *Довільна мова, породжена граматикою, яка задовольняє умову I, – індуктивна множина.*

Покажемо справедливість наведеного твердження. Для індуктивного означення потрібно визначити базис і індуктивні правила. Наступне індуктивне означення ілюструє побудову цих об'єктів. Нехай σ – аксіома (початковий нетермінал) граматики G.

Індуктивне означення мови $L(G)$

Базис: Довільне слово w, яке виводиться із σ, без застосування рекурсивних або непрямо рекурсивних правил, належить до $L(G)$.

Індукція: якщо $w \in L(G)$ і виведення $\sigma \to^* w$ включає нетермінал із рекурсивного або непрямо рекурсивного правила, то модифікуємо виведення використанням правила для побудови нового виведення $\sigma \to x$ і додаємо x до $L(G)$.

Доведення. Нехай G – граматика і M мова, яка отримана за наведеним індуктивним означенням. Покажемо, що $L(G) = M$. Зрозуміло, що $M \subseteq L(G)$, оскільки всі слова в M виводяться з аксіоми граматики G. Припустимо супротивне, тобто що $L(G) \neq M$. Це означає, що $L(G) \setminus M \neq \varnothing$. Оскільки із σ виводяться всі слова $L(G) \setminus M$, то існує деякий рядок $w \in L(G) \setminus M$, який має найкоротше лівостороннє виведення серед усіх елементів із $L(G) \setminus M$. Можна також припустити, що це виведення використовує рекурсивне або непрямо рекурсивне правило B граматики G. В протилежному випадку рядок $w \in M$, оскільки його можна було б віднести до базисних елементів у індуктивному означенні, а це суперечить тому, що $w \in L(G) \setminus M$. Відтак, лівостороннє виведення w набуває вигляду, в якому s і t термінальні слова, а α, β і γ є проміжними формами, які не включають B:

$$\sigma \to^* sB\gamma \to^* stB\beta\gamma \to st\alpha\beta\gamma \to^* w.$$

Отже, можна замінити $sB\gamma \to^* stB\beta\gamma$ в цьому виведенні на $sB\gamma \to \to s\alpha\gamma$ і отримати таке виведення термінального слова u:

$$\sigma \to^* sB\gamma \to s\alpha\gamma \to^* u.$$

Отримане виведення коротше ніж виведення w. А це говорить про те, що $u \in M$. Тепер, застосовуючи індукцію нашого індуктивного означення до вищенаведеного виведення слова u, отримуємо виведення слова w. А це говорить про те, що і $w \in M$. Дістаємо суперечність із припущенням, що $w \notin M$. Отже, $L(G) = M$. ■

Наслідок 3.2. *Регулярні і контекстно вільні мови – індуктивні множини.*

Доведення випливає з того, що граматики автоматної і контекстно вільної мов задовольняють умову I. Тоді, на підставі теореми 3.12 дістаємо індуктивність цих мов. ■

Приклад 3.6.2. Нехай задано таку граматику G:

$$\sigma \to e \mid aB,$$
$$B \to b \mid bB.$$

Потрібно дати індуктивне означення мови $L(G)$. Базисними елементами на підставі індуктивного означення будуть термінальні слова $\sigma \to e$ і $\sigma \to aB \to ab$. Отже,

Базис: $e, ab \in L(G)$.

Знайдемо тепер індуктивну частину означення. Рекурсивним у G є тільки одне правило – $B \to bB$. Тоді довільний елемент із $L(G)$, виведення якого включає нетермінал B, повинно мати вигляд $\sigma \to aB \to^+ ay$ для деякого слова y. Отже, можна використати правило $B \to bB$ для того, щоб додати один крок до виведення, тобто

$$\sigma \to aB \to abB \to^+ aby.$$

А це дає індуктивне правило для означення $L(G)$:

Індукція: якщо $ay \in L(G)$, то додати aby в $L(G)$.

З $ab \in L(G)$ випливає, що $abb \in L(G)$. Використовуючи $abb \in L(G)$, дістаємо що $abbb \in L(G)$ і т. д. Можна висловити гіпотезу, що $L(G) = \{e\} \cup \{ab^n \mid n \in N\}$. ♠

Оскільки мови, які тут розглянуто, є індуктивними множинами, то до них можна застосовувати теоретико-множинні операції. Це дає можливість будувати складніші мови з існуючих простіших мов. Тоді виникає питання: як побудувати граматику отриманої таким способом мови за заданими граматиками простих мов?

Зосередимося на трьох операціях побудови: об'єднання, добутку й ітерації мов.

Нехай L і S – мови, граматики яких мають різні множини нетермінальних символів (в протилежному випадку спільні нетермінали можна перейменувати). Нехай також аксіомами мов L і S є відповідно σ_L і σ_S. Тоді отримуємо такі мови і граматики цих мов:

- **Правило об'єднання:** Мова $L \cup S$ має аксіомою σ і правило
$$\sigma \to \sigma_L \mid \sigma_S.$$
- **Правило добутку:** Мова $L \cdot S$ має аксіомою σ і правило
$$\sigma \to \sigma_L \sigma_S.$$
- **Правило ітерації:** Мова L^* має аксіомою σ і правило
$$\sigma \to \sigma_L \sigma \mid e.$$

Після побудови граматики для результуючої мови, цю граматику можна спростити за відповідними правилами.

Зауважимо, що вищенаведені правила побудови граматик, які застосовуються до граматик, що задовольняють умову I, знову дають граматики, що задовольняють умову I.

Приклад 3.6.3. Нехай $L = \{a^n \mid n \in N\}$ і $S = \{b^n \mid n \in N\}$. Мови L і S мають граматики:
$$\sigma_L \to e \mid a\sigma_L \ i \ \sigma_S \to e \mid b\sigma_S.$$
Тоді мова $L \cup S$ матиме граматику:

$\sigma \to \sigma_L \mid \sigma_S$	правило об'єднання,
$\sigma_L \to e \mid a\sigma_L$	граматика мови L,
$\sigma_S \to e \mid b\sigma_S$	граматика мови S.

Мова $L \cdot S$ матиме граматику:

$\sigma \to \sigma_L \sigma_S$	правило добутку,
$\sigma_L \to e \mid a\sigma_L$	граматика мови L,
$\sigma_S \to e \mid b\sigma_S$	граматика мови S.

Мова L^* матиме граматику:

$\sigma \to \sigma_L \sigma \mid e$	правило ітерації,
$\sigma_L \to e \mid a\sigma_L$	граматика мови L.

Приклад 3.6.4. (*Скінченні раціональні числа*) Побудуємо граматику раціональних чисел, які зображуються скінченними десятковими дробами. Тобто, побудуємо граматику, яка породжує скінченні числа вигляду $m.n$ або $-m.n$, де m і n – десяткові числа. Наприклад: 0.0 представляє число 0. Нехай σ – аксіома, тоді одержимо

$$\begin{aligned}&\sigma \to D.D \mid -D.D,\\&D \to C \mid CD,\\&C \to 0 \mid 1 \mid 2 \mid 3 \mid 4 \mid 5 \mid 6 \mid 7 \mid 8 \mid 9.\end{aligned}$$

Цим завершується побудова граматики, яка породжує раціональні числа. ♠

ЗАДАЧІ ДЛЯ САМОСТІЙНОЇ РОБОТИ

1. Нехай дано мови $L = \{e, abb, b\}$ і $M = \{bba, ab, a\}$. Знайти елементи мов:
   а) $L \cdot M$; б) $M \cdot L$; в) L^2.
2. Знайти мову L, тобто розв'язати такі рівняння:
   а) $\{e, a, ab\} \cdot L = \{b, ab, ba, aba, abb, abba, \ldots\}$;
   б) $L \cdot \{a, b\} = \{a, baa, b, bab\}$;
   в) $\{a, aa, ab\} \cdot L = \{ab, aab, abb, aa, aaa, aba, \ldots\}$;
   г) $L \cdot \{e, a\} = \{e, a, b, ab, ba, aba\}$.
3. Довести, що для довільних мов L, M, N справедливі такі тотожності:

$$\begin{array}{ll}\text{а)}\ L \cdot \{e\} = \{e\} \cdot L = L; & \text{б)}\ L \cdot \varnothing = \varnothing \cdot L = \varnothing;\\ \text{в)}\ L \cdot (M \cup N) = L \cdot M \cup L \cdot N; & (M \cup N) \cdot L = M \cdot L \cup N \cdot L;\\ \text{г)}\ L \cdot (M \cap N) = L \cdot M \cap L \cdot N; & (M \cap N) \cdot L = M \cdot L \cap N \cdot L.\end{array}$$

4. Дано граматику вигляду

$$\begin{array}{lcl}\sigma & \to & D \mid D\sigma,\\ D & \to & 0|1|2|3|4|5|6|7|8|9.\end{array}$$

Знайти

а) правила, за якими отримано виведення:

$$\sigma \to D\sigma \to 7\sigma \to 7D\sigma \to 7DD\sigma \to 78D\sigma \to 780\sigma \to 780D \to 7801;$$

б) лівостороннє виведення термінала 7801;
в) правостороннє виведення термінала 7801.

5. Для граматики G знайти індуктивне означення мови $L(G)$, де
   а) $\sigma \to e \mid aa\sigma$; б) $\sigma \to a \mid aBc$, $B \to b \mid bB$.
6. Знайти граматику для таких мов:
   а) $\{bb, bbbb, bbbbbb, \ldots\}$; б) $\{a, ba, bba, bbba, \ldots\}$;
   в) $\{e, ab, abab, abab, ababab, \ldots\}$; г) $\{bb, bab, baab, baaab, \ldots\}$.

7. Рядок w^r означає запис символів рядка w у зворотному по- рядку. Нехай $A = \{a, b, c\}$ – алфавіт. Побудувати граматику мови $L = \{w \cdot w^r \mid w \in A^*\}$.

Чи задовольняє побудована граматика умову I?

8. Побудувати граматику мови термів сигнатури $\Omega = \{+, -, \cdot, /\}$, де всі операції бінарні. Чи буде ця граматика задовольняти умову I?

9. Побудувати граматику мови булевих термів сигнатури $\Omega = \{\vee, \wedge, \neg\}$. Чи буде ця граматика задовольняти умову I?

3.7. Рекурсивні означення на індуктивних множинах

З огляду на сказане, індуктивні множини дають можливість застосовувати метод математичної індукції. Область застосування методу математичної індукції можна розширити за допомогою рекурсивних (рекурентних) співвідношень, які визначаються на індуктивних множинах. Більше того, на індуктивних множинах можна визначити функції, які задовольняють задані рекурсивні співвідношення і мають важливі властивості.

Нехай M – чум з умовою мінімальності, на якій потрібно визначити функцію $f(x)$, тобто $f : M \to S$, де S – деяка допоміжна множина. Крім того, будемо вважати, що функція f задовольняє деякі рекурсивні співвідношення.

Означення 3.1. *Функція f(x) задовольняє рекурсивні співвідношення, якщо вони однозначно визначають для довільного елемента $a \in M$ значення функції f(a) за значеннями f(b) для всіх елементів b, строго менших ніж a.*

Справедлива важлива

Теорема 3.13. *Існує єдина функція f(x), визначена на всій множині M, яка задовольняє вказані рекурсивні співвідношення і набуває довільні задані значення на всіх мінімальних елементах множини M.*

Доведення. Покажемо спочатку єдиність такої функції. Припустимо, що існує дві різні функції $f(x)$ і $g(x)$ на множині A, які задовольняють нашим умовам. Нехай існує непорожна підмножина елементів x із A, для яких $f(x) \neq g(x)$. Оскільки A – непорожна множина, то ця підмножина має мінімальний елемент a. Цей елемент не може бути мінімальним для всієї множини A, тому що тоді, за умовою, на цьому елементі $f(a)$ і $g(a)$ збігалися б. Отже, існує $b < a$ такий, що $f(b) = g(b)$. За умовою теореми, рекурсивні співвідношення однозначно визначають значення наших функцій для $x = a$ по їхніх значеннях для всіх $b < a$, а це означає, що $f(a) = g(a)$. Отже, множина A є порожньою. Одержана суперечність доводить єдиність $f(x)$.

Доведемо тепер існування функції. Припустимо, що на мінімальному елементі множини A значення шуканої функції уже задано. Позначимо через P таку властивість: елемент $a \in A$ задовольняє властивості P, якщо на множині C всіх таких елементів x, що $x \leq a$, може бути визначена функція $f_a(x)$, яка задовольняє рекурсивні співвідношення і набуває задане значення на мінімальному елементі множини A.

В силу припущення, P істинне на мінімальному елементі $e \in A$. Далі, якщо елементи b і c задовольняють властивості P і $b < a$, то на підставі доведеної вище єдиності шуканої функції не на множині A, а на множині $B = \{x \in A \mid x \leq b\}$, маємо $f_b(x) = f_a(x)$.

Звідси випливає, що коли всі елементи b, строго менші за елемент a, задовольняють властивості P, то і сам елемент a задовольняє цій властивості. Одержуємо функцію $f_a(x)$, яка задовольняє всі вимоги, якщо для довільного елемента $b < a$ покласти $f_a(b) = f_b(b)$, а за $f_a(a)$ взяти те значення, яке однозначно визначається рекурсивними співвідношеннями.

На підставі умови індуктивності можна говорити, що для всіх a з A істинно $P(a)$. Покладаючи тепер $(\forall a \in A)\ f_a(a) = f(a)$, дістаємо функцію $f(x)$, яка має всі необхідні властивості. ■

Саме цією властивістю рекурсивно визначеної функції користуються в програмуванні.

У програмуванні підпрограму, яка обчислює значення деякої функції, називають **процедурою.**

Означення 3.4. *Функцію або процедуру f(x) називають рекурсивною, якщо вона визначається в термінах самої себе, тобто коли принаймні одне значення f(x) визначається через друге значення f(y), де $x \geq y$.*

Багато корисних рекурсивно визначених функцій мають області визначення, які є індуктивно визначеними множинами. Розглянемо два методи побудови рекурсивних функцій і рекурсивних процедур.

Побудова рекурсивно визначених функцій. Якщо S – індуктивно визначена множина, то побудуємо функцію f, задану на множині S, за такими правилами:

Базис: Для кожного базисного елемента $x \in S$ задаємо $f(x)$.

Індукція: Задаємо одне або кілька правил (для довільного індуктивно визначеного елемента $x \in S$), за якими обчислюється значення $f(x)$ за раніше знайденими значеннями функції f.

Подібно будують рекурсивно визначену процедуру P:

Базис: Для кожного базисного $x \in S$ задаємо множину дій $P(x)$.

Індукція: Задаємо одне або кілька правил (для довільного індуктивно визначеного елемента $x \in S$), за якими будуть визначатися дії для $P(x)$ у термінах раніше визначених дій із P.

Приклад 3.7.1. Нехай $f : \mathcal{N} \to \mathcal{N}$ – функція, для кожного $n \in \mathcal{N}$ визначає суму непарних чисел від 1 до $2n+1$, тобто

$$f(n) = 1 + 3 + \ldots + (2n+1).$$

Зауважимо, що $f(0) = 1$ і тому рекурсивне означення f потрібно визначати на базисному елементі $0 \in \mathcal{N}$. Для індуктивної частини означення знаходимо, яким способом можна записати $f(n+1)$ в термінах $f(n)$:

$$\begin{aligned} f(n+1) &= 1+3+\ldots+(2n+1)+[2(n+1)+1] \\ &= (1+3+\ldots+2n+1)+2n+3 \\ &= f(n)+2n+3. \end{aligned}$$

Дістаємо складові для рекурсивного означення функції f:

Базис: $f(0)=1$.
Індукція: $f(n+1)=f(n)+2n+3$.

Альтернативною формою рекурсивного означення функції f є умовна форма:

Базис: $f(0)=1$.
Індукція: якщо $n>0$, то $f(n)=f(n-1)+2(n-1)+3$. ♠

Обчислити значення рекурсивно визначеної функції легко методом *розгортки*. Саме так ці обчислення виконує зкомп'ютер. Наприклад, для обчислення значення $f(4)$ процес розгортається таким чином:

$$\begin{aligned} f(4) &= f(3)+2\cdot3+3=f(2)+2\cdot2+3+2\cdot3+3= \\ &= f(1)+2\cdot1+3+2\cdot2+3+2\cdot3+3 \\ &= f(0)+2\cdot0+3+2\cdot1+3+2\cdot2+3+2\cdot3+3 \\ &= 1+3+2\cdot1+3+2\cdot2+3+2\cdot3+3=1+3+5+7+9=25. \end{aligned}$$

Рекурсивні означення функцій і операцій на індуктивній множині "Список". Нехай $\mathcal{N}$ – множина натуральних чисел, X – деякий алфавіт і $p=y_1y_2\ldots y_m$ – довільне слово з X^*, тоді

1) $head(p)=y_1$ $(head: F(X)\to F(X))$.

2) $tail(p)=y_2\ldots y_m$ $(tail: F(X)\to F(X))$.

Зміст наведених нижче функцій і процедури їх реалізації випливають з означення.

3) $l: F(X)\to\mathcal{N}$ – функція довжини слова:
Базис: $l(e)=0$.
Індукція: $l(p)=1+l(tail(p))$.

4) $add: F(X)\times N\times X\to F(X)$, де $add(p,i,x)=y_1\ldots y_i\ xy_{i+1}\ldots y_m$, $0\le i\le l(p)$.
Базис: $add(e,0,x)=x$.
Індукція: якщо $i>0$ і $p\ne e$, то $add(p,i,x)=hl(p)\cdot x\cdot tr(p,i)$.

5) $del: F(X)\times N\to F(X)$, де $del(p,i)=y_1\ldots y_{i-1}y_{i+1}\ldots y_m$, $1\le i\le l(p)$.
Базис: $del(p,0)=p$.
Індукція: якщо $i>0$, то $del(p,i)=hl(p,i-1)\cdot tr(p,i+1)$.

6) $hl: F(X)\times N\to F(X)$, де $hl(p,i)=y_1\ldots y_i,\ 0\le i\le l(p)$.
Базис: $hl(p,0)=e$.

Індукція: якщо $i > 0$, то $hl(p,i) = head(p) \cdot hl(tail(p), i-1)$.

7) $tr : F(X) \times N \to F(X)$, де $tr(p,i) = y_{i+1} \dots y_m,\ 0 \leq i \leq l(p)$.

Базис: $tr(p,0) = p$.

Індукція: якщо $i > 0$, то $tr(p,i) = tr(tail(p), i-1)$.

8) $push : F(X) \times X \to F(X)$, де $push(p,x) = p \cdot x = add(p, l(p), x)$.

Базис: $push(p,e) = p$.

Індукція: $push(p,x) = p \cdot x = add(p, l(p), x)$.

(9) $pop : F(X) \to F(X)$, де $pop(p) = y_1 \dots y_{m-1} = del(p, l(p))$.

Базис: $pop(e) = e$.

Індукція: якщо $p \neq e$, то $pop(p) = del(p, l(p))$.

Всі вище побудовані індуктивні множини на підставі теорем 2.6 і 3.13 збігаються з множиною, яка визначається індуктивними правилами.

А функції і операції, визначені на індуктивних множинах, індуктивно, чи рекурсивно, єдині.

Список літератури

[1] *Polya G.* Mathematics and Plausible Reasoning. *– Princeton University Press. 1968 – v. 1. – 231 p.*

[2] *Анісімов А.В., Дорошенко А.Ю, Погорілий С.Д., Дорогий Я.Ю.* Програмування числових методів мовою PYTHON. *– ВПЦ "Київський університет". – 2015. – 639 с.*

[3] *Бородін О. І.* Теорія чисел. *– Київ:"Радянська школа". – 1965. – 261 с.*

[4] *Вишенський В.А., Перестюк М.О.* Комбінаторика: перші кроки. *– Кам'янець-Подільський: "Аксіома". – 2010.– 223 с.*

[5] *Єжов І.І., Скороход А.В., Ядренко М.Й.* Елементи комбінаторики. *– К.:"Вища школа". – 1972.– 72 с.*

[6] *Кривий С.Л.* Дискретна математика. *Чернівці-Київ: "Букрек". – 2017. – 567 с.*

[7] *Кривий С. Л.* Збірник задач з дискретної математики. *– Київ-Чернівці: "Букрек", 2018. – 455 с.*

[8] *Ляшко І.І., Боярчук О.К., Гай Я.Г., Головач Г.П.* Математичний аналіз в прикладах і задачах. т.1. *– Київ:"Вища школа". – 1974. – 679 с.*

[9] *Скрипченко Ю.А.* Математична індукція в геометрії. *– ж. Дидактика математики: проблеми і дослідження. – 2011. – випуск 35. с. 131–136.*

[10] *Henkin L.* On Mathematical Induction. *– The American Mathematical Monthly. – 1960 – v.67. – N 4. – pp. 323–338.*

[11] *Graham R.L., Knuth D.E., Patashnik O.* Concrete mathematics (sec. edition). *– Addison-Wesley. – 2018. – 691 p.*

Предметний покажчик

Навчальне видання

Кривий Сергій Лук'янович
Ченцов Олексій Ілліч

МАТЕМАТИЧНА ІНДУКЦІЯ
(основи, застосування, моделі)

Комп'ютерна верстка авторська