ЛЕКЦІЯ 2

ТОТОЖНОСТІ АЛГЕБРИ МНОЖИН
(повторення)

Розбиття і покриття.
Упорядковані множини.
Декартовий добуток множин.
Відповідності на множинах.

ТОТОЖНОСТІ АЛГЕБРИ МНОЖИН

Тотожності алгебри множин, сформульовані в наведених нижче теоремах, можуть бути перевірені шляхом формального доведення або на діаграмах Венна.

Таблиця 1

1. Комутативність об'єднання $X \cup Y = Y \cup X$	1. Комутативність перетину $X \cap Y = Y \cap X$
2. Асоціативність об'єднання $X \cup (Y \cup Z) = (X \cup Y) \cup Z$	2. Асоціативність перетину $X \cap (Y \cap Z) = (X \cap Y) \cap Z$
3. Дистрибутивність об'єднання відносно перетину $X \cup (Y \cap Z) = (X \cup Y) \cap (X \cup Z)$	3. Дистрибутивність перетину відносно об'єднання $X \cap (Y \cup Z) = (X \cap Y) \cup (X \cap Z)$
4. Закони дії з порожньою і	4. Закони дії з порожньою і

універсальною множинами $X \cup \varnothing = X$ $X \cup \bar{X} = U$; $X \cup \neg X = U$ $X \cup U = U$	універсальною множинами $X \cap U = X$ $X \cap \bar{X} = \varnothing$; $X \cap \neg X = \varnothing$ $X \cap \varnothing = \varnothing$
5.Закон ідемпотентності об'єднання Термін і**демпотентність** означає властивість математичного об'єкта, яка проявляється в тому, що повторна дія над об'єктом *не змінює* його $X \cup X = X$	5. Закон ідемпотентності перетину $X \cap X = X$
6. Закон де Моргана $\overline{X \cup Y} = \bar{X} \cap \bar{Y}$ $\neg(X \cup Y) = \neg X \cap \neg Y$	6. Закон де Моргана $\overline{X \cap Y} = \bar{X} \cup \bar{Y}$ $\neg(X \cap Y) = \neg X \cup \neg Y$

7. Закон поглинання $X \cup (X \cap Y) = X$	7. Закон поглинання $X \cap (X \bigcup Y) = X$
8. Закон склеювання $(X \cap Y) \cup (X \cap \overline{Y}) = X$ $(X \cap Y) \cup (X \cap \neg Y) = X$	8. Закон склеювання $(X \cup Y) \cap (X \cup \overline{Y}) = X$ $(X \cup Y) \cap (X \cup \neg Y) = X$
9. Закон Порецького $X \cup (\overline{X} \cap Y) = X \cup Y$ $X \cup (\neg X \cap Y) = X \cup Y$	9. Закон Порецького $X \cap (\overline{X} \cup Y) = X \cap Y$ $X \cap (\neg X \cup Y) = X \cap Y$
10. Закон подвійного доповнення $\overline{\overline{X}} = X \quad \neg\neg X = X$	

Способи доведення тотожностей

1. Для доведення можна використовувати тотожності алгебри логіки. Доведемо в такий спосіб властивість дистрибутивності множин.

ТЕОРЕМА. Для множин X і Y справедлива рівність

$$X \cap (Y \cup Z) = (X \cap Y) \cup (X \cap Z)$$

ДОВЕДЕННЯ. При доведенні скористаємося операціями алгебри логіки, оскільки для доведення співвідношень на множинах необхідно довести, що ці співвідношення справедливі для всіх його елементів.

$x \in X \cap (Y \cup Z) \leftrightarrow (x \in X) \wedge (x \in (Y \cup Z)) \leftrightarrow$ Визначення перетину

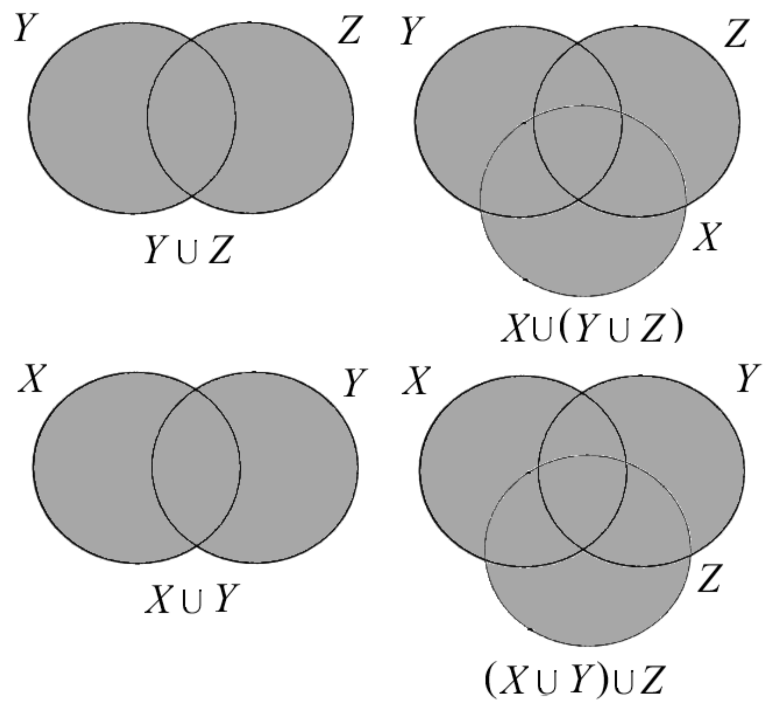
$\leftrightarrow (x \in X) \wedge ((x \in Y) \vee (x \in Z)) \leftrightarrow$ Визначення об'єднання

$\leftrightarrow ((x \in X) \wedge (x \in Y)) \vee ((x \in X) \wedge (x \in Z)) \leftrightarrow$ Дистрибутивний закон

$\leftrightarrow (x \in (X \cap Y)) \vee (x \in (X \cap Z)) \leftrightarrow$ Визначення перетину

$\leftrightarrow x \in ((X \cap Y) \cup (X \cap Z))$ Визначення об'єднання

2. Для доведення також використовують діаграми Венна (Ейлера) Доведемо властивість асоціативності за допомогою діаграм Венна



1. Будуємо $(Y \cup Z)$ і потім $X \cup (Y \cup Z)$
2. Будуємо $(X \cup Y)$ і потім $(X \cup Y) \cup Z$

3. Для доведення тотожностей алгебри множин можна використовувати інші тотожності алгебри множин.

ТЕОРЕМА. Для множин X і Y справедлива рівність

$$(X \cap Y) \cup (X \cap \overline{Y}) = X$$ - закон склеювання

ДОВЕДЕННЯ. Доведемо цю тотожність двома способами: аналітично (використовуючи тотожності алгебри множин) і конструктивно (використовуючи діаграми Ейлера-Венна).

$(X \cap Y) \cup (X \cap \overline{Y}) =$ початковий вираз

$= \left(X \cup (X \cap \overline{Y})\right) \cap \left(Y \cup (X \cap \overline{Y})\right) =$ застосували закон дистрибутивності відносно $(X \cap \overline{Y})$

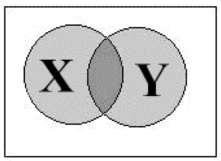
$= \left(X \cup (X \cap \overline{Y})\right) \cap (Y \cup X) =$ застосували закон Порецького

$= X \cap (Y \cup X) =$ застосували закон поглинання для об'єднання

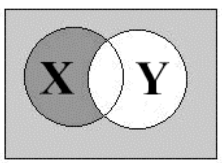
$= X$ застосували закон поглинання для перетину

Для доказу закону склеювання можна використовувати діаграми Ейлера-Венна

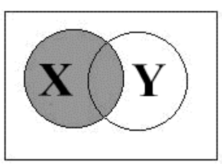
$$\left(X \cap Y\right) \cup \left(X \cap \overline{Y}\right) = X$$ – закон склеювання



$$X \cap Y$$



$$X \cap \overline{Y}$$



$$\left(X \cap Y\right) \cup \left(X \cap \overline{Y}\right) = X$$

Приклад. Доведемо тотожність: $A\setminus(B\cup C)=(A\setminus B)\setminus C$

Доведення:

1) Доведемо, що $A\setminus(B\cup C)\subseteq(A\setminus B)\setminus C$.

Розглянемо довільний елемент множини $A\setminus(B\cup C)$:

$x\in A\setminus(B\cup C)\Rightarrow x\in A$ і $x\notin B\cup C\Rightarrow x\in A$ і $x\notin B$ і $x\notin C\Rightarrow x\in A\setminus B$ й $x\notin C\Rightarrow x\in(A\setminus B)\setminus C$.

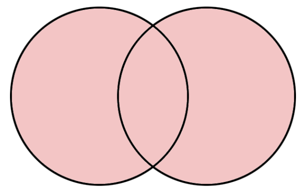
2) Доведемо, що $(A\setminus B)\setminus C\subseteq A\setminus(B\cup C)$.

Нехай $x\in(A\setminus B)\setminus C$:

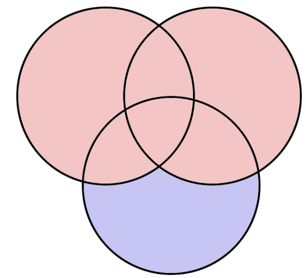
$x\in(A\setminus B)\setminus C\Rightarrow x\in A\setminus B$ і $x\notin C\Rightarrow x\in A$ і $x\notin B$ і $x\notin C\Rightarrow x\in A$ й $x\notin B\cup C\Rightarrow$

$\Rightarrow x\in A\setminus(B\cup C)$.

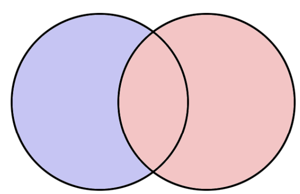
Зобразимо обидві частини тотожності за допомогою кіл Ейлера-Венна:



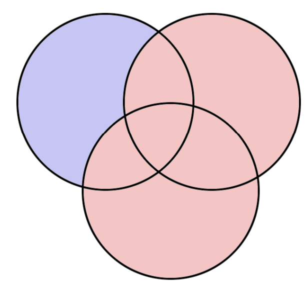
$$\left(B \cup C\right)$$



$$A \setminus \left(B \cup C\right)$$



$$\left(A \setminus B\right)$$



$$\left(A \setminus B\right) \setminus C$$

Розбиття множини

Множина X може бути розбита на класи підмножин X_j, що не перетинаються, якщо:

- об'єднання всіх підмножин X_j збігається із множиною X:

$$X = \bigcup_{j \in J} X_j$$

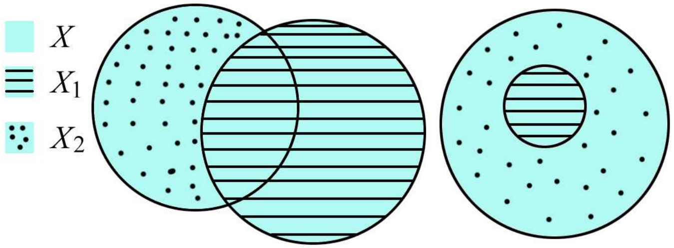
- перетин двох різних підмножин порожній,
тобто для будь-яких двох $i \in J$ і $j \in J$ при $i \neq j$ виконується умова:

$$X_i \cap X_j = \varnothing.$$

Приклад розбиття множини

1. Довільна множина X розбивається на дві підмножини X_1 та $X_2 = X \setminus X_1$, які доповнюють одна одну за умови, що

$$X_1 \cup X_2 = X \text{ та } X_1 \cap X_2 = \varnothing$$



2. Множину двозначних чисел $X = \{10, 11, 12, \ldots, 98.99\}$ можна розбити на класи за ознакою остачі від ділення на 4:

клас, породжений остачею 0 - $X_0 = \{12, 16, 20, \ldots, 96\}$;

клас, породжений остачею 1 - $X_1 = \{13, 17, 21, \ldots, 97\}$;

клас, породжений остачею 2 - $X_2 = \{10, 14, 18, \ldots, 98\}$;

клас, породжений остачею 3 - $X_3 = \{11, 15, 19, \ldots, 99\}$.

Покриття множини

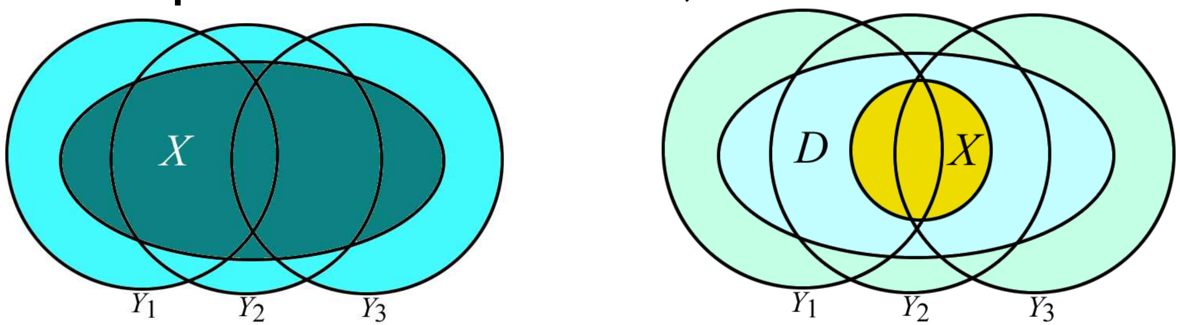
Покриттям множини X називається сімейство множин

$$C = \left\{ Y_j \right\}_{j \in J}$$

таких, що їх об'єднання містить множину X:

$$X \subset \bigcup_{j \in J} Y_j$$

Якщо C — покриття множини X,



то будь-яку множину $D \subset C$,
яка також є покриттям множини X,
називають **підпокриттям** множини C.

Приклад покриття множини

Нехай

$$X = \left\{ i \middle| i = 2n + 1, n = 0,1,2,\ldots. \right\},$$

$$J = \{1,2\}, C = \{Y_j\}_{j \in J} = \{Y_1, Y_2\},$$

$$Y_1 = \left\{ -k \middle| k = 1,2,\ldots \right\}, \qquad Y_2 = \left\{ k \middle| k = 0,1,2,\ldots \right\}.$$

Тоді

$$X \subset Y_1 \cup Y_2,$$

а, отже, сімейство множин C є покриттям множини X.

Упорядкований набір

Упорядкованим *набором (або кортежем) називають таку сукупність елементів, у послідовності яких кожний елемент займає певне місце.*

Самі елементи при цьому називаються компонентами кортежу.

Приклади:

1) Множина людей, що стоять у черзі;
2) множина букв у слові;
3) числа, що виражають довготу й широту точки на місцевості;
4) координатна пара (або трійка) в аналітичній геометрії.

Число елементів кортежу називають його ***довжиною***.

Таким чином, корте́ж або ***n*-ка** (**упорядкована *n*-ка**) — упорядкований скінченний набір елементів довжини *n* (де *n* — будь-яке натуральне число або 0). Кожний з елементів набору $x_i, 1 \le i \le n$ належить деякій множині X.

Для позначення впорядкованого набору (або кортежу) використовують

круглі дужки !!!!!

(на відміну від фігурних дужок для позначення довільної множини):

$X_1 = (x_1, x_2, ..., x_n)$-кортеж, $X_2 = \{x_1, x_2, ..., x_n\}$-множина

Відповідно до визначення, кортежі з довжиною 2 називають парами або впорядкованими парами, Кортежі з довжиною 3 - трійками, 4 - четвірками і т. ін.

Окремі випадки кортежу:

1) (x_1) кортеж з одного елемента;

2) $(\ )$ порожній кортеж, тобто кортеж з кількістю елементів 0.

На відміну від довільної множини, елементи кортежу можуть повторюватися. Багато математичних об'єктів формально визначаються як кортежі.

Наприклад.

1. Орієнтований граф визначається як кортеж (*V*,*E*), де *V* — це набір вершин, а *E* — підмножина *V* × *V*, що позначає ребра.

2. Точка в n-вимірному просторі дійсних чисел визначається як кортеж довжини n, який складається з елементів множини дійсних чисел.

Упорядкована пара

Упорядкована пара $\left(a,b\right)$ – часто вживаний математичний об'єкт.

Основна її властивість – **єдиність**.

Ця властивість виражається у наступному:

якщо $\left(a,b\right)$ і $\left(x,y\right)$ – упорядковані пари і

стверджують, що $\left(a,b\right)=\left(x,y\right)$, то $a=x$ і $b=y$.

Упорядкована множина

Будь-яку множину можна зробити впорядкованою, якщо, наприклад, переписати всі елементи множини в деякий список

Наприклад: $\left\{b, a, c, \ldots\right\} \Rightarrow \left(a, b, c, \ldots\right)$,

а потім поставити у відповідність кожному елементу номер місця, на якому він розміщений в списку. Очевидно, що кожну множину, яка містить більше одного елемента, можна впорядкувати не єдиним способом. Упорядковані множини вважаються різними, якщо вони відрізняються або своїми елементами, або їх порядком.

Наприклад: $\left\{b, a, c, \ldots\right\} \Rightarrow \left(\ldots, c, b, a,\right)$

Перестановки впорядкованої множини

Різні впорядковані множини, що відрізняються лише порядком елементів (тобто можуть бути отримані з тієї ж самої множини), називаються **перестановками** цієї множини.

Число різних способів, якими може бути впорядкована дана множина, дорівнює числу перестановок цієї множини. Число перестановок множини з n елементів дорівнює

$$P_n = n!$$

Приклад перестановки впорядкованої множини

Нехай дана неупорядкована множина

$$X=\left\{a,b,c\right\},\left|X\right|=3,P_3=3!=6.$$

Перестановки мають вигляд:

$$\underset{1\ 2\ 3}{\overset{1}{(a,b,c)}},\underset{1\ 3\ 2}{\overset{2}{(a,c,b)}},\underset{2\ 1\ 3}{\overset{3}{(b,a,c)}},\underset{2\ 3\ 1}{\overset{4}{(b,c,a)}},\underset{3\ 1\ 2}{\overset{5}{(c,a,b)}},\underset{3\ 2\ 1}{\overset{6}{(c,b,a)}}.$$

Алгоритм упорядкування множини

Нехай дано неупорядковану множину

$$A = \{a_1, a_2, a_3, \ldots, a_n\},$$

Елементами множини A є (цілі) числа. Часто в програмуванні потрібно впорядкувати елементи множини A, наприклад, у порядку зростання значень цих чисел. Таку операцію називають *сортуванням*, а множину A визначають *як масив.*

«Швидке сортування» (Quicksort)

У наведеному алгоритмі використані наступні позначення:

`a[k]` - масив чисел, у якому проводиться сортування.

Процедура дозволяє сортувати довільну підмножину масиву `a[k]`

`g` – номер мінімального елемента масиву, з якого починається сортування.

`r` – номер максимального елемента масиву, на якому закінчується сортування.

g 5>4				2<4 r	
5	3	4	1	2	a[k]
$i=1$		4		$j=5$	
2	3	4	1	5	$i<j$
	3<4	$4 \not< 4$	1<4		
2	3	1	4	5	$i<j$
2	3	1	4	5	
2<3	$3 \not< 3$	1<3			
2	1	3	4	5	$i<j$
2	1	3	4	5	
$2 \not< 2$	1<3				
1	2	3	4	5	$i<j$

1. На кожній ітерації виділяють частину масиву чисел — робочий масив.

На першій ітерації – це весь масив чисел a[k], у якому проводиться сортування. Встановлюємо границі робочого масиву $g = 1$ і $r = n$.

3. Вибирають елемент x:=a[(g+r)div2], який розміщений посередині робочого масиву.

4. Далі, починаючи з $i = 1$, послідовно збільшуємо значення i на одиницю й порівнюємо кожний елемент a_i з x, поки не буде знайдено елемент a_i такий, що $a_i > x$.

4. Потім, починаючи з $j = r$, послідовно зменшуємо значення j на одиницю, поки не буде знайдений елемент a_j такий, що $x > a_j$.

5. Якщо для знайдених елементів a_i і a_j виконується умова $i \leq j$, то ці елементи в робочому масиві міняються місцями.

6. Описана процедура закінчиться знаходженням головного елемента й поділом масиву на дві частини: одна частина буде містити елементи, менші за x, а інша — більші за x.

$\leqslant x$	x	$x \geqslant$

Далі для кожної такої частини знову застосовується описана вище процедура. Паскаль-Програма, що реалізує даний алгоритм для 10-елементного масиву, має такий вигляд:

```
Program Qsort;
Const N=10;
var
 a:array[1..N] of integer; (* вихідний масив *)
 k:integer;
procedure Quicksort(g,r:integer);
(* Процедура швидкого сортування*)
var i,j,x,y: integer;
begin
 i:=g; j:=r; x:= a[(g+r) div 2];
 repeat
  while (a[i]<x) do inc(i);
  while (x<a[j]) do dec(j);
  if (i<=j) then
  begin
   y:=a[i]; a[i]:=a[j]; a[j]:=y; inc(i); dec(j);
  end;
 until (i>j);
(*Рекурсивне використання процедури Quicksort *)
if (g<j) then Quicksort(g,j);
if (i<r) then Quicksort(i,r);
end;
begin
  writeln('Уведіть',N, 'елементів масиву:') ;
   for k:=1 to N do readln(a[k]);
  Quicksort(1,N);
  writeln('Після сортування:');
   for до:=1 to N do write(a[k],' ');
end.
```

1. $a=\{5,3,4,1,2\}$ Quicksort(1,5)
2. $x=a\left[\left(1+5\right)\operatorname{div}2\right]=a[3]=4\ \ i=1,\ j=5$
3. $5\not<4\rightarrow i=1,\quad 4\not<2\rightarrow j=5$
4. $i\le j\rightarrow 1<5\rightarrow a=\{2,3,4,1,5\},\ \ i=2,\ j=4$
5. $3<4\rightarrow \operatorname{inc} i\rightarrow i=3,\quad 4\not<4\rightarrow i=3,\ \ 4\not<1\rightarrow j=4$
6. $i\le j\rightarrow 3<4\rightarrow a=\{2,3,1,4,5\},\ \ i=4,\ j=3$

---

7. $i>j\rightarrow 4>3\rightarrow$ Quicksort(1,3)

---

8. $x=a\left[\left(1+3\right)\operatorname{div}2\right]=a[2]=3,\ i=1,\ j=3$
9. $2<3\rightarrow \operatorname{inc} i\rightarrow i=2, 3\not<3\rightarrow i=2,\ 3\not<1\rightarrow j=3$
10. $i\le j\rightarrow 2<3\rightarrow a=\{2,1,3,4,5\},\ \ i=3,\ j=2$

---

11. $i>j\rightarrow 3>2\rightarrow$ Quicksort(1,2)

---

12. $x=a\left[\left(1+2\right)\operatorname{div}2\right]=a[1]=2,\ i=1,\ j=2$
13. $2\not<2\rightarrow i=1,\ \ 2\not<1\rightarrow j=2$
14. $i\le j\rightarrow 1<2\rightarrow a=\{1,2,3,4,5\},\ \ i=2,\ j=1$

---

15. $q=1,\ j=1, 1\not<1;\quad i=2,\ r=2,\ 2\not<2,$

Декартовий добуток множин

Декартовим (прямим) добутком множин A і B називають множина $C = A \times B$, що складається з усіх упорядкованих пар $\left(a, b\right)$ таких, що $a \in A, b \in B$, тобто

$$C = A \times B = \left\{\left(a, b\right)\middle| a \in A, b \in B\right\}.$$

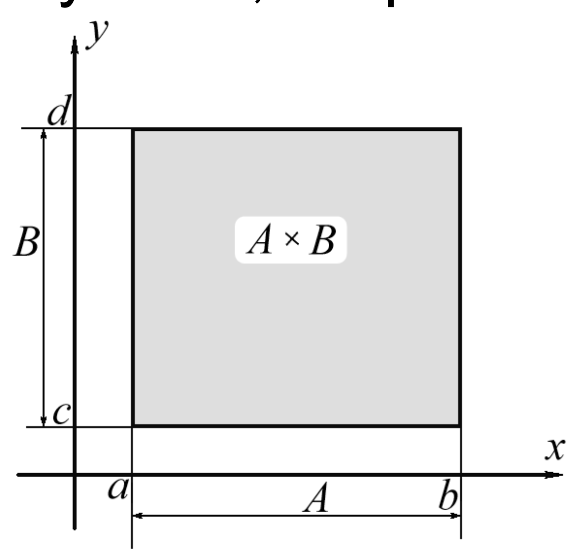
Приклад декартового добутку множин

Нехай $A = \left\{x, y, z\right\}$, $B = \left\{1, 2\right\}$.

Тоді $C = \left\{\left(x, 1\right), \left(x, 2\right), \left(y, 1\right), \left(y, 2\right), \left(z, 1\right), \left(z2\right)\right\}$.

Графічна інтерпретація декартового добутку

Існує графічна інтерпретація прямого добутку множин. Нехай множина $A = \left\{ x \middle| a \le x \le b \right\}$ – це інтервал значень змінної x і $B = \left\{ y \middle| c \le y \le d \right\}$ – це інтервал значень y . Ясно, що множини A і B мають нескінченне число елементів. Тоді прямий декартовий добуток $A \times B$ – це множина точок прямокутника, зображеного на рисунку.



Отже, $C = A \times B = \Big\{ \big(x, y \big) \Big| x \in A, y \in B \Big\}$.

У випадку декартового добутку декількох множин використовуємо показник степеня:

$A \times A = A^2$, $A \times A \times A = A^3$, $\underbrace{A \times A \times A \times ... \times A}_{n} = A^n$.

Таким чином, *n*=2,3,…

Однак таке представлення добутку множини може бути розширене на $A^1 = A, A^0 = \big\{ \Lambda \big\}$, де Λ – проекція кортежу на порожню множину осей, тобто порожній кортеж.

Зворотний декартовий добуток

Нехай $C = A \times B$ – прямий декартовий добуток множин.

Тоді $C^{-1} = B \times A$ будемо називати **зворотним** декартовым добутком до прямого добутку C.

Проектування кортежу і його графічна інтерпретація

У математиці прийнято позначати через R множину дійсних чисел. Тоді $R^2 = R \times R$ є площина дійсних чисел, а $R^3 = R \times R \times R$ представляє тривимірний простір дійсних чисел.

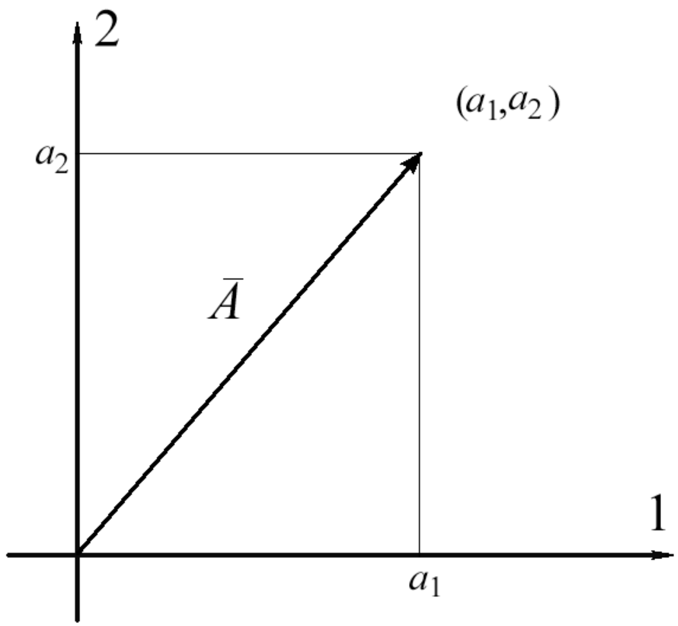
Розглянемо площину дійсних чисел або двовимірний простір дійсних чисел:

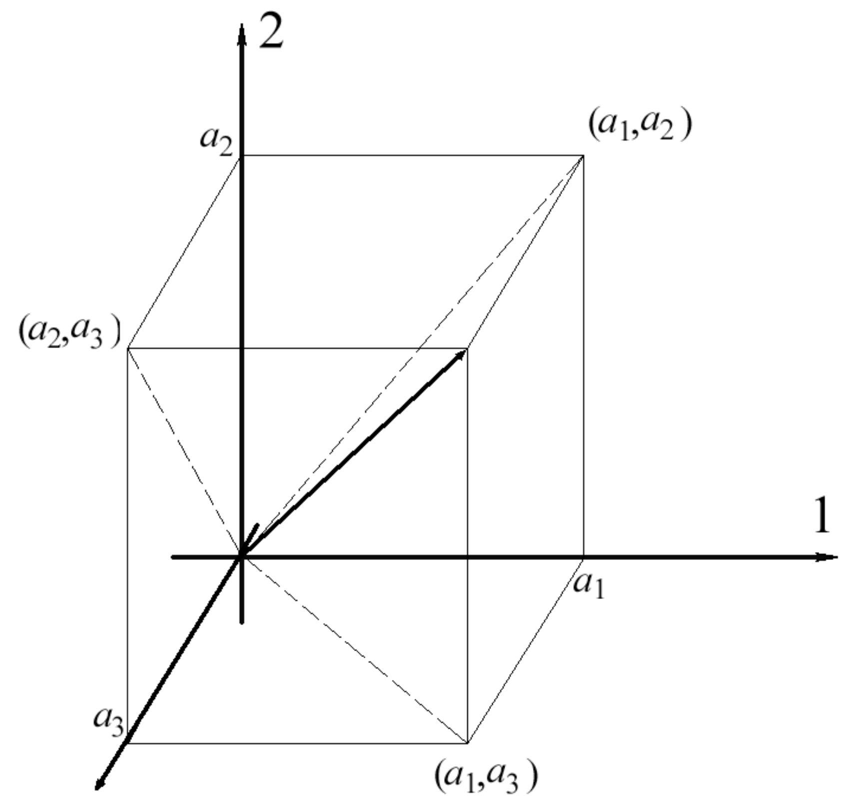
Кортеж (a_1, a_2) – це точка на площині або вектор, проведений з початку координат у дану точку. Компоненти a_1 і a_2 – це **проекції** вектора $\overline{A} = (a_1, a_2)$ на осі 1 і 2. Цей факт скорочено записують так:

$$proj_1 \overline{A} = proj_1(a_1, a_2) = a_1,$$

$$proj_2 \overline{A} = proj_2(a_1, a_2) = a_2.$$

Кортеж $\left(a_1, a_2, a_3\right)$ – це точка в тривимірному просторі або тривимірний вектор, проведений з початку координат у точку з координатами $\left(a_1, a_2, a_3\right)$.





Проекції вектора на осі координат у цьому випадку записуються так:

$$proj_1\overline{A} = proj_1\left(a_1, a_2, a_3\right) = a_1,$$
$$proj_2\overline{A} = proj_2\left(a_1, a_2, a_3\right) = a_2,$$
$$proj_3\overline{A} = proj_3\left(a_1, a_2, a_3\right) = a_3.$$

Проектувати можна не тільки на осі, але й на координатну площину. У цьому випадку проекція є двоелементним кортежем:

$$proj_{1,2}\overline{A} = proj_{1,2}\left(a_1, a_2, a_3\right) = \left(a_1, a_2\right),$$
$$proj_{1,3}\overline{A} = proj_{1,3}\left(a_1, a_2, a_3\right) = \left(a_1, a_3\right),$$
$$proj_{2,3}\overline{A} = proj_{2,3}\left(a_1, a_2, a_3\right) = \left(a_2, a_3\right).$$

Узагальнюючи поняття проекції на n-вимірний простір, можна n-елементну впорядковану множину $\left(a_1, a_2, a_3, \ldots, a_n\right)$ розглядати як точку в n-вимірному просторі. У цьому випадку

$$proj_i \overline{A} = proj_i \left(a_1, a_2, a_3, ..., a_i, ..., a_n \right) = a_i,$$

$$proj_{i,j} \overline{A} = proj_{i,j} \left(a_1, a_2, a_3, ..., a_i, ..., a_j, ..., a_n \right) = \left(a_i, a_j \right),$$

$$proj_{i,j,k} \overline{A} =$$

$$= proj_{i,j,k} \left(a_1, a_2, a_3, ..., a_i, ..., a_j, ..., a_k, ..., a_n \right) = \left(a_i, a_j, a_k \right)'$$

....

Очевидно, що загальна кількість осей, на які припустиме проектування, дорівнює $n-1$.

Нехай множина D складається з кортежів довжини m. Тоді проекцію множини D називають множину проекцій кортежів з D.

Приклад:

$$D = \left\{ \left(1,2,3,4,5\right), \left(3,2,1,5,4\right), \left(2,3,6,7,1\right), \left(8,1,1,4,6\right) \right\}.$$

$$D = \{(1,2,3,4,5),(3,2,1,5,4),(2,3,6,7,1),(8,1,1,4,6)\}.$$

Проектування кортежів на одну вісь:

$$proj_1 D = \{(1),(3),(2),(8)\},$$

$$proj_2 D = \{(2),(2),(3),(1)\},$$

$$proj_3 D = \{(3),(1),(6),(1)\},$$

$$proj_4 D = \{(4),(5),(7),(4)\},$$

$$proj_5 D = \{(5),(4),(7),(6)\}.$$

Проектування кортежів на дві осі:

$$proj_{1,2} D = \{(1,2),(3,2),(2,3),(8,1)\},$$

$$proj_{1,3} D = \{(1,3),(3,1),(2,6),(8,1)\},$$

…………………………………………

$$proj_{2,3}D = \{(2,3),(2,1),(3,6),(1,1)\},$$

$$proj_{1,3}D = \{(1,3),(3,1),(2,6),(8,1)\},$$

…………………………………………

Проектування кортежів на три осі:

$$proj_{1,2,3}D = \{(1,2,3),(3,2,1),(2,3,6),(8,1,1)\}$$

…………………………………………..

$$proj_{3,4,5}D = \{(3,4,5),(1,5,4),(6,7,7),(1,4,6)\}$$

…………………………………………

Операція проектування може застосовуватися тільки до тих множин, які містять кортежі однакової довжини.

Відповідності і відношення на множинах

Відповідність. Основні поняття

Дано множини X і Y. (студенти і виробники мобілок)

Елементи цих множин можуть зіставлятися один з одним яким-небудь чином, утворюючи впорядковані пари

$$\left(x, y\right).$$

Якщо спосіб зіставлення визначений, тобто для кожного елемента $x \in X$ вказано елемент $y \in Y$, з яким зіставляється елемент x, то говорять, що між множинами X та Y установлена відповідність.

Для того, щоб задати відповідність, необхідно вказати:

1) множину X, елементи якої зіставляються з елементами іншої множини;

2) множину Y, елементи якої ставлять у відповідність елементам першої множини;

3) множину $Q \subseteq X \times Y$, що визначає закон (правило), за яким здійснюється відповідність, тобто таке правило, що перераховує всі пари (x, y), що беруть участь у зіставленні.

Таким чином, **відповідність** (позначимо її через q) **є трійкою множин**

$$q = \left\langle X, Y, Q \right\rangle$$

де

$Q \subseteq X \times Y$ – підмножина декартового добутку множин X і Y, яку ще називають графіком відповідності;

X – множина відправлення відповідності;

Y – множина прибуття відповідності;

Область визначення та область значень

Крім того, з кожною відповідністю зв'язані нерозривно ще дві множини:

1. множина $proj_x Q$, яку називають **областю визначення** відповідності. В цю множину входять елементи множини X, що беруть участь у зіставленні;

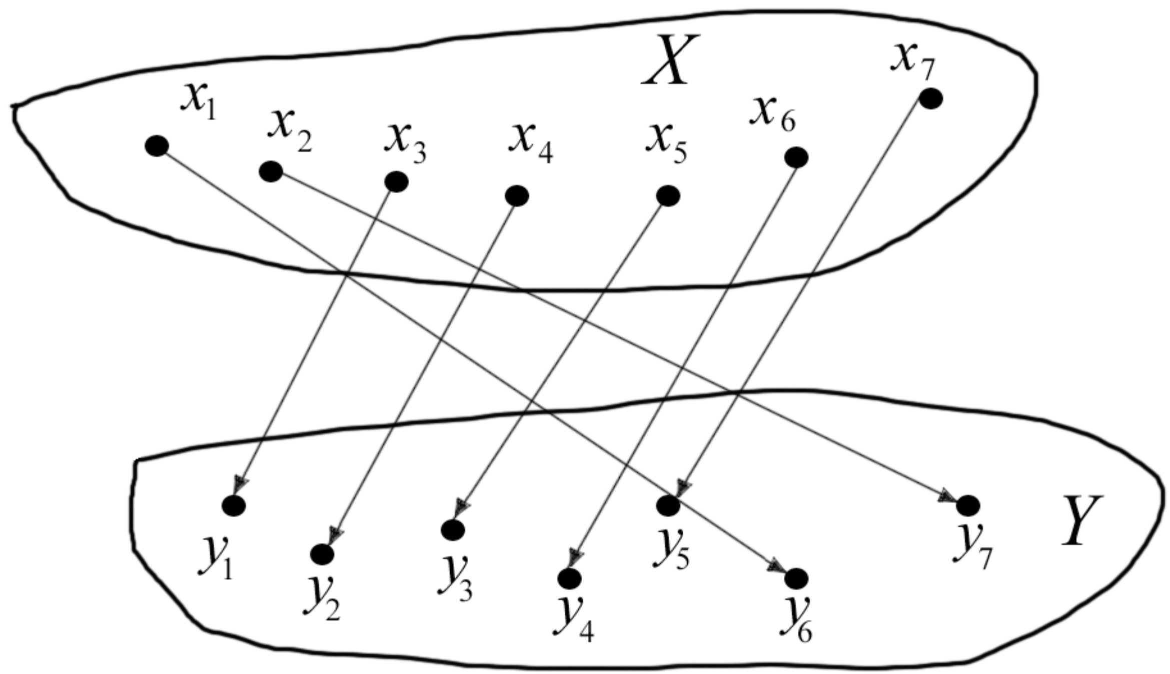
2. множина $proj_y Q$, яку називають **областю значень** відповідності. В цю множину входять елементи множини Y, що беруть участь у зіставленні.

Якщо $(x, y) \in Q$, то говорять, що елемент y відповідає елементу x. Геометрично це зображується у вигляді стрілки, спрямованої від x до y:

На рисунку показано дві множини X і Y з установленими відповідностями між їх елементами.

При цьому графік відповідності має вигляд:

$$Q = \left\{ \left(x_1, y_6 \right), \left(x_2, y_7 \right), \left(x_3 . y_1 \right), \left(x_4, y_2 \right), \left(x_5, y_3 \right), \left(x_6, y_4 \right), \left(x_7, y_5 \right) \right\}$$



Зворотна відповідність

Для кожної відповідності

$$q = \left\langle X, Y, Q \right\rangle,\ Q \subseteq X \times Y$$

існує зворотна відповідність, яка утворюється у випадку, коли дану відповідність розглядають у зворотному напрямку, тобто визначають елементи $x \in X$, з якими зіставляються елементи $y \in Y$.

Зворотна відповідність позначається:

$$q^{-1} = \left\langle X, Y, Q^{-1} \right\rangle, \text{ де } Q^{-1} = Y \times X.$$

Геометрично зворотну відповідність можна одержати шляхом зміни напрямку стрілок прямої відповідності.

Типи відповідностей

Відповідності можуть бути різних типів: одно-однозначні, одно-багатозначні, багато-однозначні, багато-багатозначні.

А) **Одно-однозначна** (або **взаємно-однозначна)** відповідність – це така попарна відповідність між елементами двох множин X і Y, коли з одним елементом з X зіставлено один елемент з Y і навпаки.

Приклад. Нехай існує множина натуральних чисел N і множина квадратів натуральних чисел P. Кожному натуральному числу можна поставити у відповідність його квадрат і навпаки – кожному квадрату натурального числа відповідає саме натуральне число. Тому між множинами N й P існує взаємно-однозначна відповідність.

Б) **Одно-багатозначна** відповідність – це така відповідність між елементами двох множин X і Y, коли з одним елементом першої множини X зіставлено більше одного елемента другої множини Y, але кожний елемент другої множини відповідає тільки одному елементу першої множини.

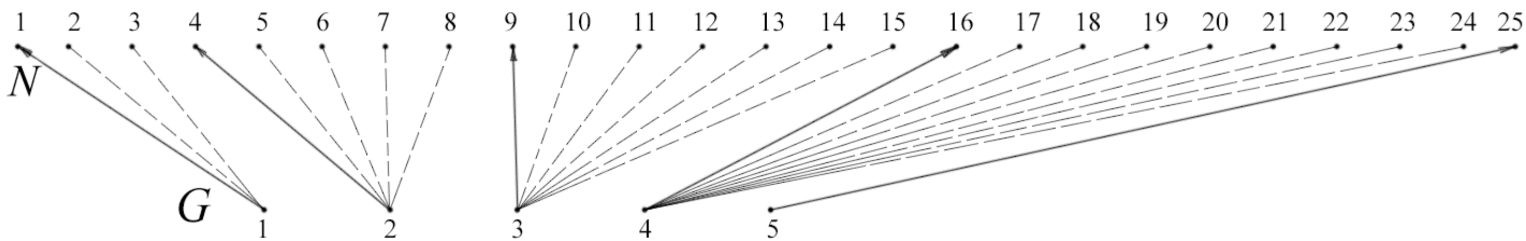
Приклад. Нехай існує:

- множина квадратних коренів з цілих чисел:
$G = \left\{1, 2, 3, 4, 5\right\}$,

- множина цілих чисел
$N = \left\{1, 2, 3, 4, 5, 6, 7, 8, 9, 10, 11, 12, 13, 15, 16, \ldots, 25\right\}$.

Кожному елементу множини G **однозначно** відповідає один елемент множини N.

Зворотна відповідність може бути **багатозначною** за умови, що будемо виділяти цілу частину від взяття квадратного кореня від кожного елемента множини N.



В) **Багато-однозначна** відповідність – це така відповідність між елементами двох множин X і Y, коли з елементом першої множини зіставлено тільки один елемент другої множини, але кожний елемент другої множини відповідає більше, ніж одному елементу першої множини.

Приклад. Нехай

$X = \{1, 2, 3, ..., 25\}$ – множина студентів у групі,
$Y = \{2, 3, 4, 5\}$ – припустима множина оцінок.

Кожний студент під час здачі іспиту може одержати тільки одну оцінку. У той час та сама оцінка може бути поставлена деякій підмножині студентів.

Г) **Багато - багатозначна** відповідність – це така відповідність між елементами двох множин X і Y, коли з одним елементом першої множини зіставлено більше одного елемента другої множини і навпаки.

Приклад.

Нехай X – множина театральних постановок, а
Y – множина глядачів.

Кожний глядач може подивитися **деяку підмножину** театральних постановок.

У той же час, кожну з театральних постановок відвідує **деяка підмножина** глядачів.