

ЛЕКЦІЯ 5

Відношення порядку

Відношення порядку

Існують відношення, що визначають порядок розташування елементів множини.

1. Умова відношення, коли елементами множини T є стани динамічної системи:

« $t_i < t_j$ »- t_j **слідуює за** t_i

або « $t_j > t_i$ »- t_i **передуює** t_j

$$T = \{t_0, t_1, t_2, t_3, \dots, t_{n-1}\},$$

де $t_0 < t_1 < t_2 < t_3 < \dots < t_{n-1}$.

Символи «<» «>» використовують для порівняння величин відрізків часу, вимірюваних від початку відліку.

Запишемо ці відношення **у вигляді предиката**:

$$R_1 = \{(t_i, t_j) \mid t_i < t_j \text{ при } i < j\} \quad R_2 = \{(t_j, t_i) \mid t_j > t_i \text{ при } j > i\}$$



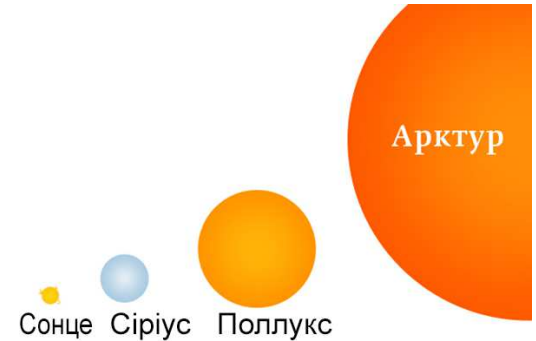
2. Умова відношення, коли елементами множини A є числа або об'єкти, що мають властивість виражену

числом: $A = \{a_0, a_1, a_2, \dots, a_i, \dots, a_j, \dots, a_n\}$,

« $a_j > a_i$ »- a_j > **більше** a_i

або « $a_i < a_j$ »- a_i **менше** a_j

де $a_0 < a_1 < a_2 < \dots < a_i < \dots < a_j < \dots < a_n$.



Символами «>» або «<» користуються для порівняння чисел.

Відношення R у вигляді предиката, задане на $A \times A$.

$$R_1 = \left\{ (a_i, a_j) \mid a_i < a_j \text{ при } i < j \right\}$$

$$R_2 = \left\{ (a_j, a_i) \mid a_j > a_i \text{ при } j > i \right\}$$

3. Умова відношення, коли елементами множини A є множини:

$$A = \{A_0, A_1, \dots, A_i, \dots, A_j, \dots, A_n\}.$$

« $A_i \subseteq A_j$ » - A_i **ВХОДИТЬ В** A_j ,

де

$$A_0 \subseteq A_1 \subseteq \dots \subseteq A_i \subseteq \dots \subseteq A_j \subseteq \dots \subseteq A_n$$

або

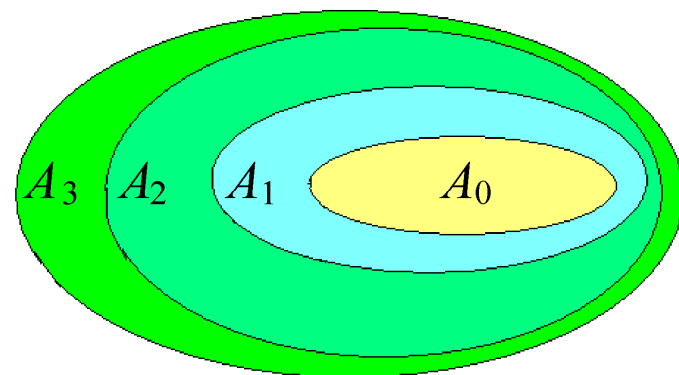
« $A_i \subset A_j$ » - A_i **СТРОГО ВХОДИТЬ В** A_j

$$A_0 \subset A_1 \subset \dots \subset A_i \subset \dots \subset A_j \subset \dots \subset A_n.$$

Відношення R у вигляді предиката, задане на $A \times A$.

$$R_1 = \left\{ (A_i, A_j) \mid A_i \subseteq A_j \text{ при } i < j \right\} \quad R_2 = \left\{ (A_i, A_j) \mid A_i \subset A_j \text{ при } i < j \right\}$$

У всіх випадках можна розташувати елементи множин у деякому порядку або, інакше кажучи, ввести відношення порядку на множині.



Визначення відношень порядку

Відношення порядку на множині A поділяють на:

- відношення строгого порядку;
- відношення нестроного порядку.

Визначення 1. Відношення R називають **відношенням строгого порядку** на множині A , якщо воно має властивості:

- *антирефлексивності*, тобто якщо xRy то $x \neq y$.
- *антисиметричності*, тобто, якщо xRy і yRx , то $x = y$.
- *транзитивності*, тобто, якщо xRy і yRz , то xRz .

Визначення 2. Відношення R називають **відношенням нестроного порядку** на множині A , якщо воно має властивості:

- *рефлексивності*, тобто, xRx .
- *антисиметричності*, тобто, якщо xRy і yRx , то $x = y$.
- *транзитивності*, тобто, якщо xRy і yRz , то xRz .

Термінологія та позначення

1. **Відношення нестрогого порядку** позначають символом « \leq » за аналогією з відношенням «менше або дорівнює» на множині дійсних чисел. При цьому, якщо $a \leq b$, те говорять, що елемент a **не перевищує** b або елемент a **підпорядкований** b .

2. **Відношення строгого порядку**. Якщо $a \leq b$ і $a \neq b$, то пишуть $a < b$ і говорять, що елемент a **менший** b або, елемент a **строго підпорядкований** b .

3. **Загальний випадок відношень**. Іноді, щоб відрізнити відношення порядку на деякій множині від відношення порядку на множині дійсних чисел, використовують спеціальні символи

Відношення порядку на множинах « \preceq » і « \prec » «зводять до».

Відношення порядку на числах « \leq » і « $<$ » «менше за»

Приклад. Відношення порядку в \mathbb{R}^n

1. Відношення нестрогого порядку для кортежів:

$$(a_1, \dots, a_{i-1}, a_i, a_{i+1}, \dots, a_n) \leq (b_1, \dots, b_{i-1}, b_i, b_{i+1}, \dots, b_n)$$

справедливе за умови, що

$$a_1 \leq b_1, \dots, a_{i-1} \leq b_{i-1}, a_i \leq b_i, a_{i+1} \leq b_{i+1}, \dots, a_n \leq b_n$$

2. Відношення строгого порядку для кортежів:

$$(a_1, \dots, a_{i-1}, a_i, a_{i+1}, \dots, a_n) < (b_1, \dots, b_{i-1}, b_i, b_{i+1}, \dots, b_n),$$

справедливе за умови, що

$$a_1 < b_1, \dots, a_{i-1} < b_{i-1}, a_i < b_i, a_{i+1} < b_{i+1}, \dots, a_n < b_n$$

Однак для встановлення нестрогого порядку достатньо, щоб умова $a_i \leq b_i$ була виконана хоча б по одній координаті, тобто

$$(a_1, \dots, a_{i-1}, a_i, a_{i+1}, \dots, a_n) \leq (b_1, \dots, b_{i-1}, b_i, b_{i+1}, \dots, b_n),$$

якщо

$$a_1 < b_1, \dots, a_{i-1} < b_{i-1}, a_i \leq b_i, a_{i+1} < b_{i+1}, \dots, a_n < b_n$$

Приклад. Відношення порядку в \mathbb{R}^2

$$(a_1, a_2) \leq (b_1, b_2)$$

Це відношення справедливе, якщо $a_1 \leq b_1$ і $a_2 \leq b_2$

$(1, 5) \leq (1, 7) \rightarrow 1 = 1$ і $5 < 7$; $(1, 5)$ і $(5, 1)$ **непорівнювані**

$$(a_1, a_2) < (b_1, b_2)$$

$(1, 5) < (2, 7) \rightarrow (1 < 2)$ і $5 < 7$; $(1, 5)$ і $(7, 2)$ **непорівнювані**

Приклад. Відношення порядку в \mathbb{R}^3

$$(a_1, a_2, a_3) \leq (b_1, b_2, b_3)$$

Це відношення справедливе, якщо $a_1 \leq b_1$, $a_2 \leq b_2$ і $a_3 \leq b_3$

$(1, 2, 3) \leq (1, 2, 4) \rightarrow 1 = 1$, $2 = 2$, $3 < 4$

$$(a_1, a_2, a_3) < (b_1, b_2, b_3)$$

Це відношення справедливе, якщо $a_1 < b_1$, $a_2 < b_2$ і $a_3 < b_3$

$(1, 2, 3) < (2, 3, 4) \rightarrow 1 < 2$, $2 < 3$, $3 < 4$

Види відношень порядку:

1. Строгий повний порядок

Відношення строгого порядку **задане на всіх** елементах упорядкованої множини.

2. Строгий частковий порядок

Відношення строгого порядку **задане не на всіх елементах** упорядкованої множини.

3. Нестрогий повний порядок

Відношення нестрогого порядку **задане на всіх** елементах упорядкованої множини.

4. Нестрогий частковий порядок

Відношення нестрогого порядку **задане не на всіх елементах** упорядкованої множини.

Основні поняття про впорядковані множини

Упорядковані множини утворюють один з фундаментальних типів математичних структур.

Визначення впорядкованої множини

(множина + відношення порядку)

Упорядкованою множиною називають непусту множину X разом із заданим на ньому бінарним нестрогим відношенням порядку « \leq », яке за визначенням:

- 1) **рефлексивне**: $a \leq a$;
- 2) **антисиметричне**: $a \leq b \leq a \Rightarrow a = b$ (для будь-яких a, b, X).
- 3) **транзитивне**: $a \leq b \leq c \Rightarrow a \leq c$;

або строгим « $<$ » відношенням порядку, яке за визначенням:

- 1) **антирефлексивне**: $a < b \Rightarrow a \neq b$;
- 2) **антисиметричне**: $a < b \wedge b < a \Rightarrow a = b$
- 3) **транзитивне**: $a \leq b \leq c \Rightarrow a \leq c$;

Визначення порівнюваності упорядкованих множин.
Елементи a і b упорядкованої множини називають *порівнюваними*, якщо $a < b$, $a = b$ або $a > b$. Знаки $<$, $=$ і $>$ мають звичайний зміст.

Визначення лінійно впорядкованої множини (ланцюга).

1. (через властивість порівнюваності)

Упорядкована множина X називається *лінійно впорядкованою*, або *ланцюгом*, якщо будь-які два її елементи **порівнювані**.

2. (через відношення лінійного порядку)

Упорядковану множину X називають *лінійно впорядкованою*, або *ланцюгом*, якщо на ній задане **відношення лінійного порядку**, тобто $\forall a, b : (aRb \vee bRa)$

Відношення лінійного порядку — це відношення, яке має властивість **рефлексивності**.

Такий порядок завжди повний та нестрогий.

Ланцюг та антиланцюг

1.Ланцюг —лінійно впорядкована множина.

2.Ланцюг —лінійно впорядкована підмножина частково впорядкованої множини

Приклади лінійно впорядкованих множин (ланцюгів)

Натуральні числа — найменша лінійно впорядкована множина, що не має верхньої межі.

$$A = \{1, 2, 3, 4, \dots\}$$

Цілі числа — найменша лінійно впорядкована множина, що не має ні верхньої, ні нижньої межі.

$$A = \{\dots - 4, -3, -2, -1, 0, 1, 2, 3, 4, \dots\}$$

Антиланцюг — упорядкована множина, у якій жодні два різні елементи не є порівнюваними.

$$A = \{a, 1, \square\}$$

Властивості лінійно впорядкованих множин

1. Покриття

Нехай X – довільний ланцюг. Якщо $a < b$ в X і не існує елемента $c \in X$ з умовою $a < c < b$ (розташованого між a і b), то співвідношення $a < b$ називають *покриттям*.

Приклад. $X = \{a, b, c, d, e, f\}$.

Покриття: $a < b, b < c, c < d, d < e, e < f$

Не є покриттями: $a < c$, оскільки $a < b < c$; $b < d$, оскільки $b < c < d$

2. Взаємне положення елементів: $X = \{a, b, c, d, e, f\}$

Елемент a називають *попереднім* для b

Елемент b називають *наступним за* a .

Елемент ланцюга, у якого немає попереднього елемента, називають *граничним елементом*. a – граничний елемент.

3. Щільний ланцюг

Ланцюг називають *щільним*, якщо в ньому немає покриттів. У щільних ланцюгах між будь-якими елементами $a < b$ лежить нескінченна кількість елементів.

Приклад. Множина дійсних чисел

Нехай $A = \{1, 1.4, 2.345, 7.87653\}$

Не є покриттями: $1 < 1.4$, оскільки $1 < 1.2 < 1.4$;
 $1.4 < 2.345$, оскільки $1.4 < 2 < 2.345$ і т.д.

4. Повний зверху ланцюг

Ланцюг називають *повним зверху*, якщо його довільна непуста підмножина має \sup (супремум).

5. Повний знизу ланцюг

Ланцюг називають *повним знизу*, якщо його довільна непуста підмножина має \inf (інфімум).

6. Повний ланцюг

Ланцюг називають *повним*, якщо він повний зверху і знизу одночасно.

Цілком упорядкована множина

Найважливіший клас ланцюгів утворюють цілком упорядковані множини

Визначення.

Ланцюг називають *цілком упорядкованою множиною*, якщо будь-яка його непуста підмножина має найменший елемент.

Приклад 1 . Ланцюг усіх натуральних чисел N

$$N = \{1, 2, 3, 4, \dots\}$$

є цілком упорядкованою множиною, оскільки її найменший елемент – 1.

Приклад 2. Усі скінченні ланцюги є прикладами цілком упорядкованих множин.

Приклад 3. Будь-яка непуста підмножина цілком упорядкованої множини цілком упорядкована.

Частково впорядкована множина

Визначення відношення часткового порядку.

Бінарне відношення R на множині X називають відношенням **часткового порядку**, якщо для деяких $a \in X$, $b \in X$ не виконується ні відношення aRb , ні відношення bRa .

Визначення частково впорядкованої множини.

Упорядковану множину називають частково впорядкованою, якщо на ній задане відношення часткового порядку.

Властивості частково впорядкованих множин (строгий та нестрогий порядок)

Визначення для нестрогого порядку.

Відношення R на X є відношенням *нестрогого часткового порядку*, якщо воно має властивості:

рефлексивності - $\forall a (aRa)$,

антисиметричності - $\forall a, b (aRb) \wedge (bRa) \Rightarrow a = b$

транзитивності - $\forall a, b, c (aRb) \wedge (bRc) \Rightarrow aRc$.

Визначення для строгого порядку.

Відношення R на X є відношенням *строого часткового порядку*, якщо воно має властивості:

антирефлексивності - $\forall a, b (aRb) \Rightarrow a \neq b$,

антисиметричності - $\forall a, b (aRb) \wedge (bRa) \Rightarrow a = b$

транзитивності - $\forall a, b, c (aRb) \wedge (bRc) \Rightarrow aRc$.

Приклад частково впорядкованої множини

Приклад 1. Нехай задано:

1. Множина T додатних дільників числа 30.

$$T = \{1, 2, 3, 5, 6, 10, 15, 30\}.$$

2. Відношення « \leq », згідно якого m і n є порівнюваними:
 $m \leq n$ за умови, що n ділиться на m націло.

Нехай $n=15$ і $m=5$. Тоді n і m – є порівнюваними, оскільки 15 ділиться на 5 націло.

Нехай $n=6$ і $m=5$. Тоді n і m – непорівнювані, оскільки 6 не ділиться на 5 націло.

Висновок.

1. Задане відношення порядку « \leq » на множині T є відношенням часткового порядку.
2. Множина T є частково упорядкованою множиною на заданому відношенні.

Розбиття частково впорядкованої множини на ланцюзі

Нехай є деяка множина A . Говорять, що множина A розбита на підмножини $A_1, A_2, A_3, \dots, A_m$, якщо:

1. $A_i \neq \emptyset$, $(i = 1, 2, \dots, m)$;
2. $A_i \cap A_j = \emptyset$, якщо $i \neq j$ для всіх $i, j \in \{1, 2, 3, \dots, m\}$;
3. $A = \bigcup_{i=1}^m A_i$

Нехай A є частково впорядкованою множиною.

Розбиття множини A на ланцюзі **називають найменшим, якщо воно має найменше число елементів m** у порівнянні з іншими розбиттями A на ланцюзі.

Таке розбиття також називають *мінімальним ланцюговим розбиттям (МЛР)* множини A .

Приклад мінімального ланцюгового розбиття (МЛР)

Нехай дана множина A :

$$A = \{1, a, \angle, б, 2, 7, в, \triangleleft, 1245, \square, \partial\},$$

на якій задані такі відношення часткового порядку:

$$R_1 = \{(a, b) \mid a \leq b\}$$

$$R_2 = \{(a, b) \mid "a \text{ слідує в алфавітному порядку за } b"\}$$

$$R_3 = \{(a, b) \mid "a \text{ має більше вуглів, ніж } b"\}$$

Побудуємо розбиття цієї множини на ланцюзі

$$A_1 = \{1, 2\}; A_2 = \{7, 1245\}; A_3 = \{a, б\}; A_4 = \{в, \partial\}; A_5 = \{\angle, \triangleleft, \square\}$$

$$m = 5 \quad A = \bigcup_{i=1}^5 A_i, A_i \neq \emptyset (i = 1, \dots, 5),$$

$$A_i \cap A_j = \emptyset \text{ якщо } i \neq j \text{ для всіх } i, j \in \{1, 2, 3, 4, 5\}.$$

$$m = 3 \quad A_1 = \{1, 2, 7, 1245\}; A_2 = \{a, б, в, \partial\}; A_3 = \{\angle, \triangleleft, \square\}$$

Отже, розбиття A_1, A_2, A_3 - МЛР

Визначення найбільшого елемента множини

Найбільшим елементом лінійно впорядкованої множини X відносно строгого « $<$ » або нестроогого « \leq » упорядкування будемо називати такий елемент $a \in X$, що для будь-якого $x \in X$ вірно $x < a$ або $x \leq a$.

Теорема про єдиність найбільшого елемента.

Якщо існує найбільший елемент лінійно впорядкованої множини, то він є єдиним.

Доведення.

Припустимо a -**найбільший елемент** і a' – **також найбільший елемент**.

Тоді для будь-якого x виконується $x \leq a$ і $x \leq a'$.

Зокрема, $a \leq a'$ або $a' \leq a$.

За властивістю **антисиметричності**, то з $(aRa') \wedge (a'Ra)$ слідує $a = a'$.

Оскільки $a = a'$, то якщо існує **найбільший елемент**, то він **єдиний**.

Тому, якщо говорять про найбільший елемент множини, то мають на увазі **цілком визначений** її елемент.

Приклад. Необхідно знайти найбільший елемент лінійно впорядкованої множини $X = \{1, 2, 15, 18\}$, заданої на відношенні нестрогого порядку $a \leq b$.

Згідно з визначенням:

1. **Усі** елементи даної множини **повинні бути меншими** або дорівнювати найбільшому.
2. Найбільший елемент – **єдиний**.

Порівняємо елементи множини X :

- 1) $1 \geq 1, 1 \not\geq 2, 1 \not\geq 15, 1 \not\geq 18$.
- 2) $2 \geq 1, 2 \geq 2, 2 \not\geq 15, 2 \not\geq 18$.
- 3) $15 \geq 1, 15 \geq 2, 15 \geq 15, 15 \not\geq 18$.
- 4) $18 \geq 1, 18 \geq 2, 18 \geq 15, 18 \geq 18$.

Необхідним умовам відповідає тільки елемент 18.

Визначення максимального елемента множини

Максимальним елементом частково впорядкованої множини X відносно строгого « $<$ » (нестроного « \leq ») порядку називають такий його елемент $a \in X$, для якого наявна одна із двох ситуацій:

- або $x < a$ ($x \leq a$),
- або a і x – непорівнювані.

Зауваження

На одній і тій же множині можуть бути задані **різні відношення** порядку.

За одним з них множина може бути лінійно впорядкованою, а за іншим – частково впорядкованою.

Тоді за першим відношенням будемо говорити **про найбільший елемент**, а за другим – **про максимальний**.

Визначення найменшого і мінімального елементів множини

Найменшим елементом лінійно впорядкованої множини X відносно строгого «<» (нестрогого « \leq ») впорядкування будемо називати такий елемент $a \in X$, що для всіх $x \in X$ вірно $a < x$ ($a \leq x$).

Мінімальним елементом частково впорядкованої множини X відносно строгого «<» або нестроного « \leq » впорядкування називають такий його елемент $a \in X$, для якого наявна одна із двох ситуацій:

- або $a < x$, ($a \leq x$)
- або a і x – непорівнювані.

Зауваження.

Якщо на множині існує **найменший елемент**, то він є **єдиним мінімальним**.

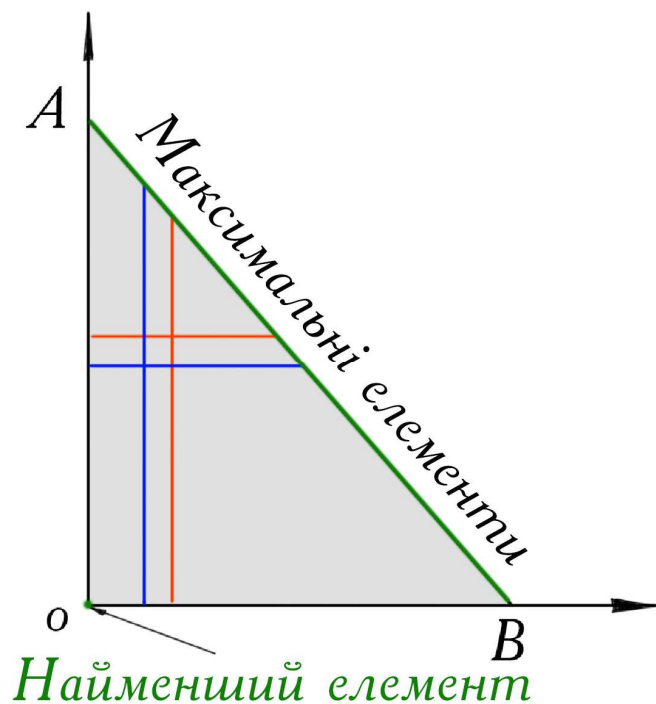
Аналогічно, якщо на множині існує **найбільший елемент**, то він є **єдиним максимальним**.

Приклад. На множині точок X , обмеженій трикутником OAB , задаємо відношенням порядку:

$(a,b) \leq (c,d)$ яке справедливе тоді і тільки тоді, коли $a \leq c$ і $b \leq d$.

$$X = \{(0,0), (0,1), (0,2), \dots, (0,A), (1,1), (1,2), \dots, (2,1), (2,2), \dots\}$$

$$X = \{(0,0), (1,0), (2,0), \dots, (A,0), (1,1), (2,1), \dots, (3,1), (4,2), \dots\}$$



Точка $(0,0)$ є найменшим елементом даної множини.

Мінімальний елемент множини X – єдиний і збігається з найменшим елементом.

Максимальними елементами множини X є всі точки, що лежать на стороні AB трикутника OAB .

Найбільший елемент множини X не існує.

Визначення верхньої і нижньої граней множини

Визначення верхньої грані

Якщо A є частково впорядкована множина і $B \subseteq A$, то елемент $a \in A$ називають **верхньою гранню** множини B , якщо для кожного $b \in B$ існує нерівність $b \leq a$.



$A = \{1, 2, 3, 4, 5, 6, 7, 8, 9\}$, $B = \{1, 2, 3, 4, 5\}$. Верхні грані: 6, 7, 8, 9

Визначення нижньої грані

Якщо A є частково впорядкована множина і $B \subseteq A$, то елемент $a \in A$ називають нижньою гранню множини B , якщо для кожного $b \in B$ існує нерівність $a \leq b$.



$A = \{1, 2, 3, 4, 5, 6, 7, 8, 9\}$, $B = \{5, 6, 7, 8, 9\}$. Нижні грані: 1, 2, 3, 4

Визначення точної верхньої грані множини

Елемент $a \in A$ називають **найменшою верхньою гранню**, якщо $a = \min_i a_i$, де a_i — верхня грань множини B .



$$A = \{1, 2, 3, 4, 5, 6, 7, 8, 9\}, B = \{1, 2, 3, 4, 5\}, a = \min \{6, 7, 8, 9\} = 6$$

Найменший елемент a множини всіх верхніх граней називають **точною верхньою гранню** або **супремумом** і позначають $\sup B$.

$$A = \{1, 2, 3, 4, 5, 6, 7, 8, 9\}, B = \{1, 2, 3, 4, 5\}. \quad \sup B = 6$$

Супремум є така верхня грань множини, яка є нижньою гранню множини всіх її верхніх граней.

Визначення точної нижньої грані множини

Елемент $a \in A$ називають **найбільшою нижньою гранню**, якщо $a = \max a_i$, де a_i – нижня грань множини B .



$$A = \{1, 2, 3, 4, 5, 6, 7, 8, 9\}, B = \{5, 6, 7, 8, 9\}. a = \max \{1, 2, 3, 4\} = 4$$

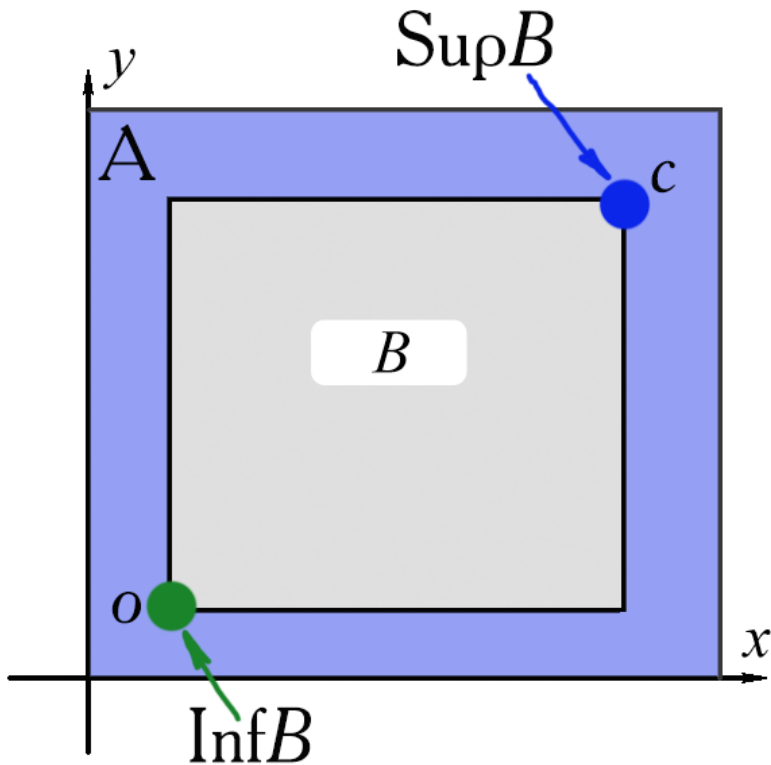
Найбільший елемент множини всіх нижніх граней називають **точною нижньою гранню** або **інфімумом** і позначають $\inf B$.

$$A = \{1, 2, 3, 4, 5, 6, 7, 8, 9\}, B = \{5, 6, 7, 8, 9\}. \inf B = 4.$$

Інфімумом є така нижня грань множини B , яка є верхньою гранню множини всіх її нижніх граней.

Приклад. Розглянемо множину A точок прямокутника із заданим відношенням порядку на підмножині B :

$(a,b) \leq (c,d)$ тоді і тільки тоді, коли $a \leq c$ і $b \leq d$.

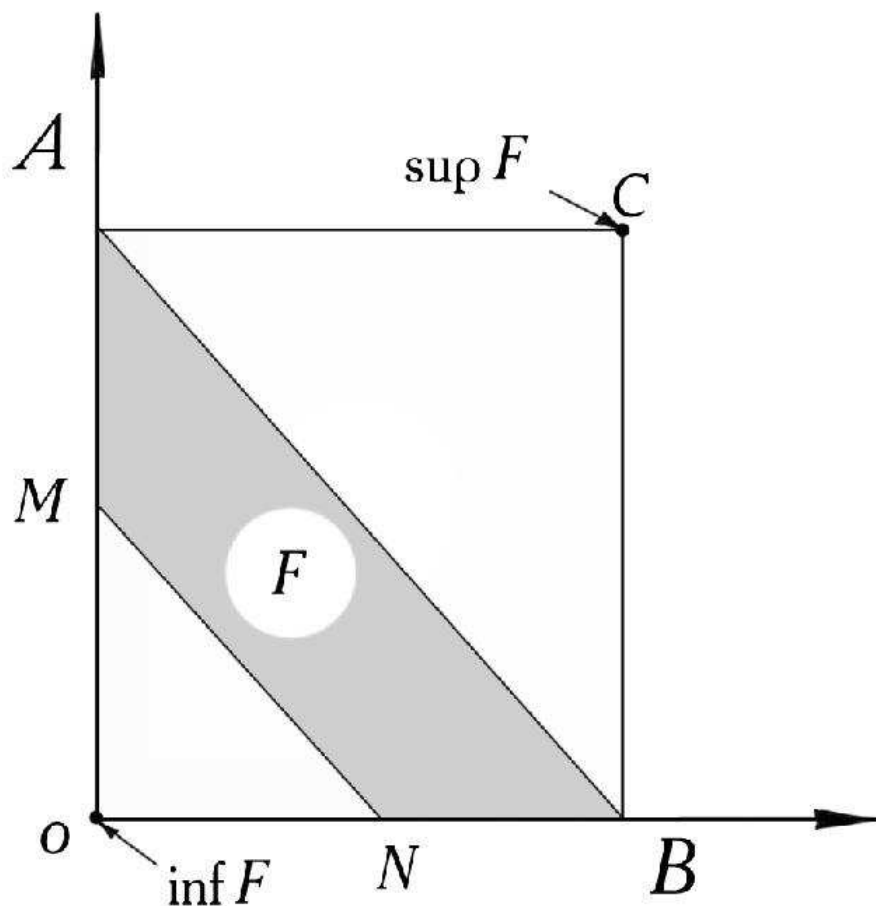


Точка o є точною
нижньою гранню $\inf B \in A$.

Точка c є точною
верхньою гранню $\sup B \in A$.

З рисунка видно, що
обидві точки належать
множині A .

Приклад. Розглянемо множину F точок трапеції $ABNM$ із заданим відношенням порядку: $(a,b) \leq (c,d)$ тоді і тільки тоді, коли $a \leq c$ і $b \leq d$.



Приклад показує,
що існує точна
верхня грань $\sup F$ і
точна нижня грань
 $\inf F$.

Однак жодна з
граней не належить
множині F .

Діаграма Хассе

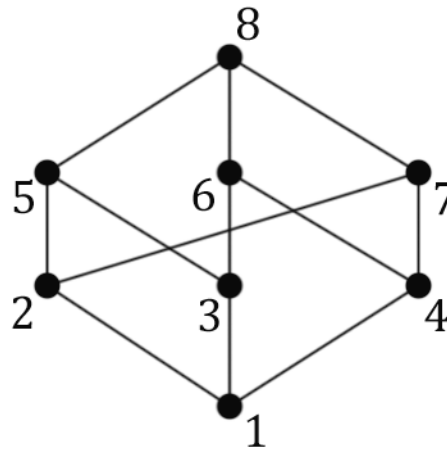
Для графічного представлення впорядкованої множини R використовують *діаграму Хассе*. Цю діаграму будують у такий спосіб.

Кожному елементу множини X ставлять у відповідність точку (кружок) на площині, причому, якщо aRb , точку, яка відповідає елементові a , розташовують нижче від точки, яка відповідає елементові b . Точки $a \in X$ і $b \in X$ з'єднують лінією (ребром), якщо aRb і не існує елемента $c \in X$ такого, що aRc й cRb .

Приклад. Нехай дана множина $X = \{1, 2, 3, 4, 5, 6, 7, 8\}$, на якій задано відношення

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

Діаграма Хассе даного відношення представлена на рисунку.



Приклад. Нехай $M = \{x, y, z\}$, а 2^M – **булеан** множини M :

$$2^M = \{\emptyset, \{x\}, \{y\}, \{z\}, \{x, y\}, \{y, z\}, \{z, x\}, \{x, y, z\}\}$$

$T \in 2^M, V \in 2^M$ -елементи булеана

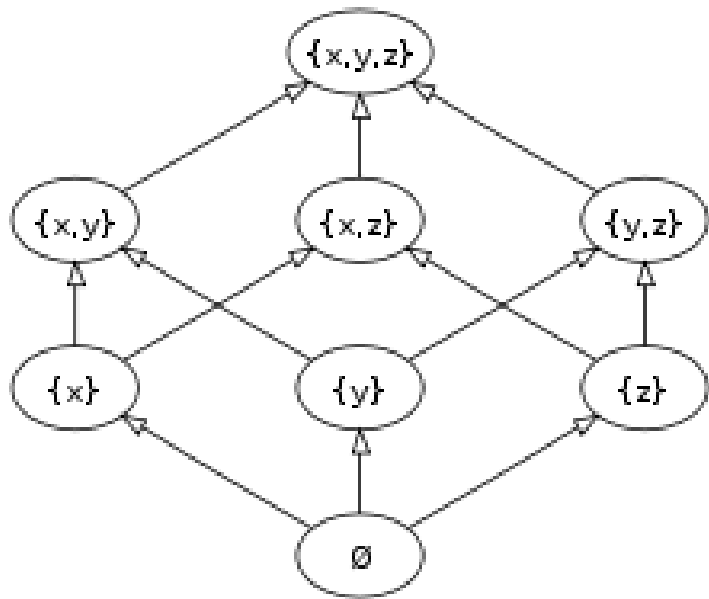
Визначимо відношення R :

$$R = \{(T, V) \mid T \subseteq V\}$$

Наприклад, $(\{y\}, \{x, y\}) \in R$,
оскільки $\{y\} \subseteq \{x, y\}$. Однак
 $(\{y, z\}, \{z\}) \notin R$, оскільки
 $\{y, z\} \not\subseteq \{z\}$.

Побудувавши відношення
 R , можна легко перевірити,

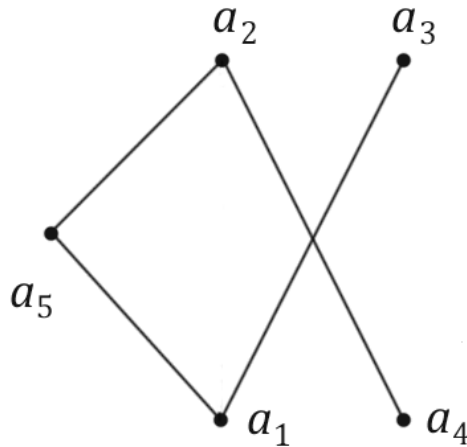
що $(2^M, R)$ — частково упорядкована множина.



Приклад.

На рисунку представлений частковий порядок, породжений бінарним відношенням

$$R = \{ (a_1, a_2), (a_1, a_3), (a_1, a_5), (a_4, a_2), (a_5, a_2) \}.$$



Діаграма Хассе допомагає краще розуміти взаємозв'язок елементів, що належать одній і тій же впорядкованій множині (наприклад, приналежність одного і того ж ланцюга або одного і того ж антиланцюга).

Приклад створення несуперечливих відношень S і R

Варіант 0. aSb якщо a сестра b і aRb якщо a дружина b .
Із двох відношень першим будемо більш строге.

aRb – більш строге в порівнянні з aSb :

- a_i – жінка;
- a_i – може входити тільки 1 раз в a_iRb_j
- b_j – чоловік;
- b_j – може входити тільки 1 раз в a_iRb_j

aSb – менш строге в порівнянні з aRb :

- a_i – жінка;
- a_i – може входити багато разів в a_iRb_j
- b_j – чоловік або жінка;
- b_j – може входити багато разів в a_iRb_j

Опис структури даних

A: Array [0..19] of String;

B: Array [0..19] of String;

масив елементів множин A и B

Gender_A: Array [0..19] of Byte;

Gender_B: Array [0..19] of Byte;

масив гендерної приналежності: 0-жінка; 1-чоловік.

Alen і Blen кількості елементів у множинах A и B.

R_relation[i, j] матриця відношення R.

S_relation[i, j] матриця відношення S.

```

procedure R_create; (*Варіант 0 – а дружина b*)
var k:integer;
    used:Boolean;
begin
    For i:=0 to Alen-1 do (* Переглядаємо A *)
    If Gender_A[i]=0 then (* Вибираємо жінку *)
    For j:=0 to Blen-1 do (* Переглядаємо B *)
    If Gender_B[j]=1 then (* Вибираємо чоловіка *)
    begin
        used:=false;
        For k:=0 to Alen-1 do (*Перевіряємо на зайнятість*)
        If R_relation[k,j]=1 then used:=true;
        If Not used then
            begin R_relation[i,j]:=1; Break; end;
    end;
end;

```

```
procedure S_create; (* Варіант 0 - а сестра b*)  
begin  
  For i:=0 to ALEN-1 do      (* Переглядаємо A *)  
    If Gender_A[i]=0 then      (* Вибираємо жінку *)  
      For j:=0 to BLEN-1 do    (* Переглядаємо B *)  
        (* Якщо жінка, то вибираємо відразу її сестрою a *)  
        (* Якщо чоловік і він не чоловік a, вибираємо його братом a *)  
        If Gender_B[j]=0 then S_Relation[i,j]:=1 else  
          If R_relation[i,j]=0 then S_Relation[i,j]:=1;  
end;
```

Варіант 1. aSb якщо a мати b і aRb якщо a внучка b .

Із двох відношень першим будуємо більш строге.

aRb – більш строге в порівнянні з aSb :

- a_i – жінка;
- a_i – може входити тільки 2 рази в a_iRb_j ;
- b_j – жінка;
- b_j – може входити багато разів в a_iRb_j

aSb – менш строге в порівнянні з aRb :

- a_i – жінка;
- a_i – може входити багато разів в a_iRb_j ;
- b_j – чоловік або жінка;
- b_j – може входити багато разів в a_iRb_j

```

procedure R_create; (*Варіант 1 – а внучка b*)
var k:integer;
    used:Byte;
begin
    For i:=0 to Alen-1 do (* Переглядаємо A *)
    If Gender_A[i]=0 then (* Вибираємо жінку *)
    For j:=0 to Blen-1 do (* Переглядаємо B *)
    If Gender_B[j]=0 then (* Вибираємо жінку *)
    begin
        used:=0;
        For k:=0 to Alen-1 do (*Тільки дві бабусі*)
        If R_relation[k,j]=1 then Inc(used);
        If used<2 then
            begin R_relation[i,j]:=1; Break; end;
        end;
    end;

```



```

procedure S_create; (*Варіант 1 – а мати b*)
var used:Boolean;
begin
    For i:=0 to Alen-1 do          (* Переглядаємо А *)
    If Gender_A[i]=0 then          (* Вибираємо жінку *)
    For j:=0 to Blen-1 do          (* Переглядаємо В *)
    begin
        used:=false;
        For k:=0 to Alen-1 do      (*b уже має матір?*)
        If s_relation[k,j]=1 then used:=true;
        If not used then
            If (Gender_B[0]=1) then
                begin S_Relation[i,j]:=1; Break; end else
                If R_relation[i,j]=0 then
                    begin S_Relation[i,j]:=1; Break; end;
            end;
    end;
end;

```

Варіант 2. aSb якщо a дружина b і aRb якщо a мати b .

aSb – більш строге в порівнянні з aRb :

- a_i – жінка;
- a_i – може входити тільки 1 раз в a_iRb_j
- b_j – чоловік;
- b_j – може входити тільки 1 раз в a_iRb_j

aRb – менш строге в порівнянні з aSb :

- a_i – жінка;
- a_i – може входити багато разів в a_iRb_j
- b_j – чоловік або жінка;
- b_j – може входити багато разів в a_iRb_j

procedure S_create; (*Варіант 2 – а дружина b*)

var k:integer;
used:Boolean;

begin

For i:=0 **to** Alen-1 **do** (* Переглядаємо A *)

If Gender_A[i]=0 **then** (* Вибираємо жінку *)

For j:=0 **to** Blen-1 **do** (* Переглядаємо B *)

If Gender_B[j]=1 **then** (* Вибираємо чоловіка *)

begin

used:=false;

For k:=0 **to** Alen-1 **do** (*Перевіряємо на зайнятість*)

If S_relation[k,j]=1 **then** used:=true;

If Not used **then**

begin S_relation[i,j]:=1; Break; **end**;

end;

end;

```

procedure R_create; (*Варіант 1 – а мати b*)
var used:Boolean;
begin
    For i:=0 to Alen-1 do          (* Переглядаємо A *)
    If Gender_A[i]=0 then          (* Вибираємо жінку *)
    For j:=0 to Blen-1 do          (* Переглядаємо B *)
    begin
        used:=false;
        For k:=0 to Alen-1 do      (*b уже має матір?*)
        If R_relation[k,j]=1 then used:=true;
        If not used then
            If (Gender_B[0]=1) then
            begin R_Relation[i,j]:=1; Break; end else
            If S_relation[i,j]=0 then
            begin R_Relation[i,j]:=1; Break; end;
        end;
    end;
end;

```

Варіант 9. aSb якщо a свекруха b , aRb якщо a батько b .

Примітка. Свекруха – мати чоловіка.

aSb – більш строге в порівнянні з aRb :

- a_i – жінка;
- a_i – може входити багато разів в a_iRb_j
- b_j – жінка;
- b_j – може входити тільки 1 раз в a_iRb_j

aRb – менш строге в порівнянні з aSb :

- a_i – чоловік;
- a_i – може входити багато разів в a_iRb_j
- b_j – чоловік або жінка;
- b_j – може входити 1 раз в a_iRb_j

```

procedure S_create; (*Варіант 9 – а свекруха b*)
(*Розглянутий варіант-у свекрухи один син*)
var k:integer;
    used:Boolean;
begin
    For i:=0 to Alen-1 do (* Переглядаємо A *)
    If Gender_A[i]=0 then (* Вибираємо жінку *)
    For j:=0 to Blen-1 do (* Переглядаємо B *)
    If Gender_B[j]=0 then (* Вибираємо жінку *)
    begin
        used:=false;
        For k:=0 to Alen-1 do (*Перевіряємо на зайнятість*)
        If S_relation[k,j]=1 then used:=true;
        If Not used then
            begin S_relation[i,j]:=1; Break; end;
    end;
end;

```

```
procedure R_create; (*Варіант 9 – а батько b*)  
var used:Boolean;  
begin  
    For i:=0 to Alen-1 do          (* Переглядаємо А *)  
    If Gender_A[i]=1 then          (* Вибираємо чоловіка *)  
    For j:=0 to Blen-1 do          (* Переглядаємо В *)  
    begin  
        used:=false;  
        For k:=0 to Alen-1 do      (*b уже має батька?*)  
        If R_relation[k,j]=1 then used:=true;  
        If not used then  
            begin R_Relation[i,j]:=1; Break; end;  
    end;  
end;
```