Санкт-Петербургский национальный исследовательский университет информационных технологий, механики и оптики

Дискретная математика

курс лекций

лекция 4

Элементы теории множеств

Кафедра «Проектирования и безопасности компьютерных систем»
Гришенцев А. Ю.
www.moveinfo.ru

Санкт-Петербург
2014

Множество

Кантор* определял множество как «*объединение в одно целое объектов, хорошо различимых нашей интуицией или мыслью*».

$$A=\{a_1, a_2, a_3, a_4, \ldots, a_n\}$$

где: $a_1, a_2, a_3, a_4, \ldots, a_n$ – элементы множества,

A – множество.

Элементы множества обычно обозначают строчными буквами, а сами множества прописными (заглавными) буквами.

*Георг Кантор - (3 марта 1845, Санкт-Петербург – 6 января 1918, Залле) – германский математик, наиболее известен как создатель теории множеств.

Способы задания множеств

Перечислением

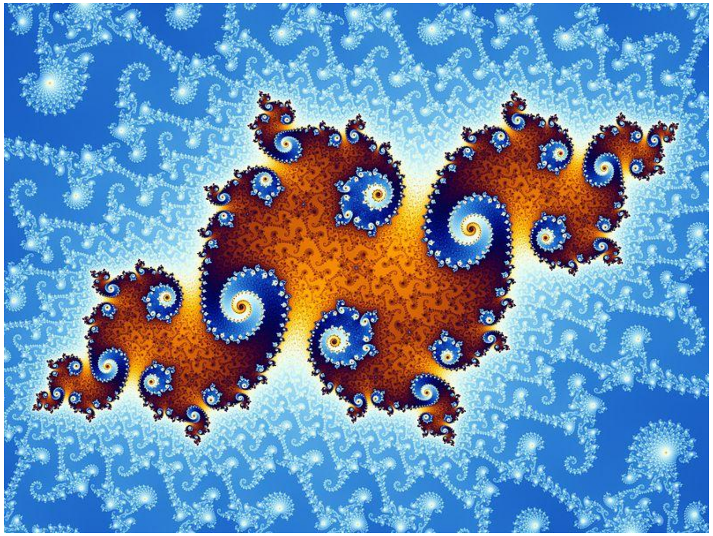
например: $A=\{a,\ b,\ c,\ d,\ e,\ f\}$

Аналитическим выражением

например: $(\forall x)(x\in \mathrm{N})\wedge(x(\mathrm{mod}\ \ 2)=1)\Rightarrow x\in X$

Графически

например:



Фракталы, (в данном случае множество Жюлиа-Мандельборта)

Множество географических объектов на глобусе

Некоторые специальные множества

$\varnothing$ – пустое множество, множество не содержащее ни одного элемента;

U – универсум, множество всех множеств содержащее все элементы;

$\mathbb{N} = \{0,1,2,\ldots\}$ – множество натуральных чисел;

$\mathbb{N}_+ = \{1,2,\ldots\}$ – множество положительных натуральных чисел;

$\mathbb{Z} = \{\ldots,-2,-1,0,1,2,\ldots\}$ – множество целых чисел;

$\mathbb{Z}_k = \mathbb{E}_k = \{0,1,2,\ldots,k-1\}$ – подмножество натуральных чисел от 0 до k-1;

$\mathbb{Q} = \{m/n,\ m,n \in \mathbb{Z},\ n \neq 0\}$ – множество рациональных чисел;

$\mathbb{R} = (-\infty,\ \infty)$ – множество вещественных чисел;

$\mathbb{C} = \{x+j\cdot y,\ x,y \in \mathbb{R},\ j^2=-1\}$ – множество комплексных чисел.

В различных научно-практических направлениях существует значительное разнообразие специальных множеств.

Операции над множествами

$\neg A = \{x: x \notin A\}$ – отрицание множества A;

$A \cup B = \{x: x \in A \vee x \in B\}$ – объединение множества A и B;

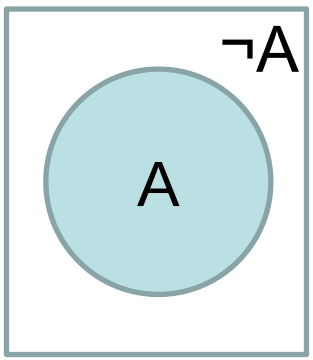
$A \cap B = \{x: x \in A \wedge x \in B\}$ – пересечение множеств A и B;

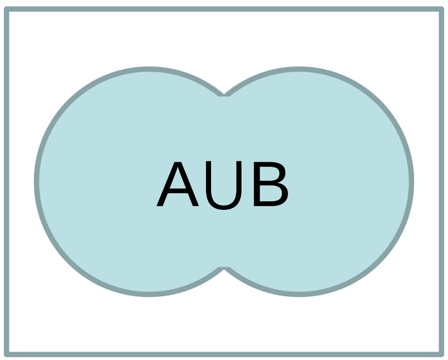
$A - B = A \setminus B = \{x: x \in A \wedge x \notin B\}$ – разность множеств A и B;

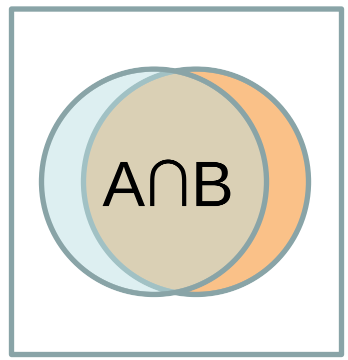
$A \dotdiv B = (A - B) \cup (B - A)$ – семетрическая разность множеств A и B;

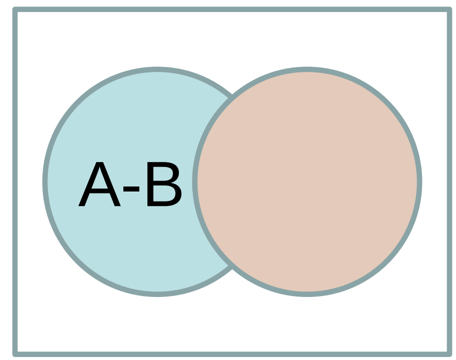
$A \times B = \{(a,b): a \in A \wedge b \in B\}$ – декартово произведение множеств A и B.

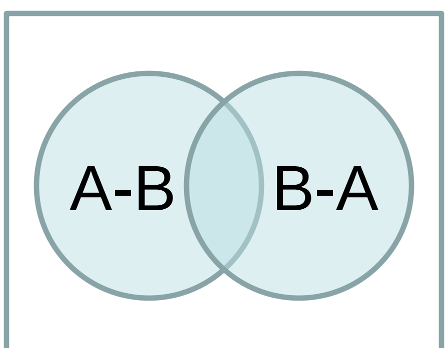
Визуализация операций на диаграммах Эйлера-Венна











Диаграммы Эйлера-Венна – это графический способ отображений операций над множествами.

Свойства операций над множествами

1. Коммутативность $A \cup B = B \cup A$, $A \cap B = B \cap A$
2. Ассоциативность $A \cup (B \cup C) = (A \cup B) \cup C$, $A \cap (B \cap C) = (A \cap B) \cap C$
3. Идемпотентность $A \cup A = A$, $A \cap A = A$
4. Правило поглощения $A \cap (A \cup B) = A$, $A \cup (A \cap B) = A$
5. Дистрибутивность

$$A \cap (B \cup C) = (A \cap B) \cup (A \cap C)\ ,\ A \cup (B \cap C) = (A \cup B) \cap (A \cup B)$$

6. Инволюция (двойное отрицание) $\neg(\neg A) = A$
7. Свойства констант $A \cap U = A$, $A \cup U = U$, $A \cap \varnothing = \varnothing$, $A \cup \varnothing = A$
8. Закон исключения третьего, закон противоречия $A \cup (\neg A) = U$, $A \cap (\neg A) = \varnothing$
9. Закон де Моргана $\neg (A \cup B) = (\neg A) \cap (\neg B)$, $\neg(A \cap B) = (\neg A) \cup (\neg B)$

Операции над множествами ассоциированы с логическими операциями. Можно сказать, что логические операции являются частным случаем операций над множествами.

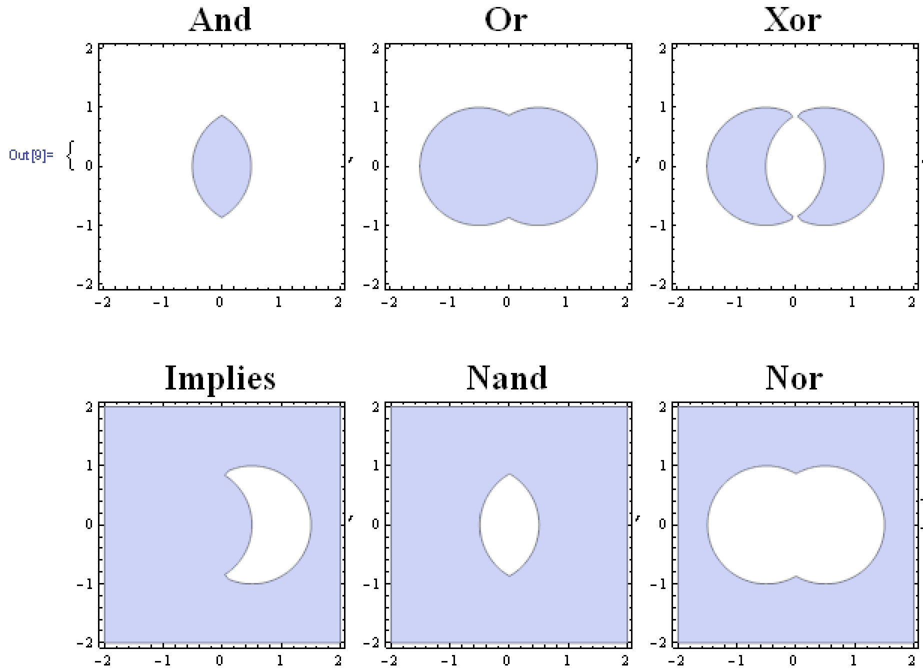
Связь операций над множествами с логическими операциями

Множества	Логика	Значения
A	**a**	**0101**
B	**b**	**0011**
Ø	false	0000
$B \cap A$	$a \land b$	0001
$B - A = \neg(B \to A)$	$\neg(b \to a)$	0010
B	b	0011
$A - B = \neg(A \to B)$	$\neg(a \to b)$	0100
A	a	0101
$A \dot{-} B = A \oplus B = \neg(A \leftrightarrow B)$	$a \oplus b = \neg(a \leftrightarrow b)$	0110
$A \cup B$	$a \lor b$	0111
$\neg(A \cup B)$	$a \downarrow b$	1000
$A \leftrightarrow B = \neg(A \dot{-} B) = \neg(A \oplus B)$	$a \leftrightarrow b = \neg(a \oplus b)$	1001
$\neg A = U - A$	$\neg a$	1010
$A \to B = \neg(A - B)$	$a \to b$	1011
$\neg B = U - B$	$\neg b$	1100
$B \to A = \neg(B - A)$	$b \to a$	1101
$\neg(A \cap B)$	$a \mid b$	1110
U	true	1111

Иллюстрация связи операций над множествами с логическими операциями при помощи пакета Mathematica

```
a=(-(1/2)+x)^2+y^2<1;
b=((1/2)+x)^2+y^2<1;
Table[RegionPlot[f[a,b],{x,-2,2},{y,-
2,2},PlotLabel→f],{f,{And,Or,Xor,Implies,Nand,Nor}}]
```

Результат выполнения программы



В пакете Mathematica запуск программы на выполнение осуществляется совместным нажатием клавиш (Shift+Enter).

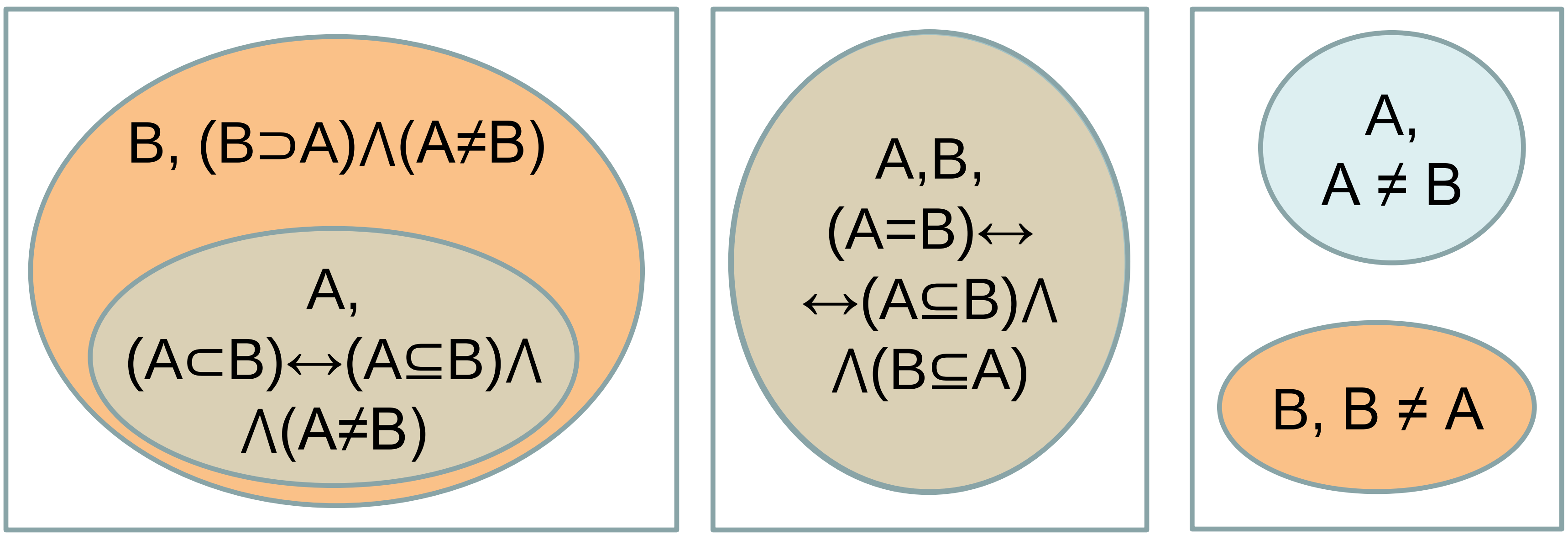
Соотношения множеств

$A \subseteq B = B \supseteq A \leftrightarrow \forall a\ (a \in A \rightarrow a \in B)$

$A = B \leftrightarrow (A \subseteq B) \wedge (B \subseteq A)$

$A \subset B = B \supset A \leftrightarrow (A \subseteq B) \wedge (A \neq B)$

На диаграммах Эйлера-Венна



Возможны обозначения:

$A \not\subset B$, $A \not\supset B$, $A \not\subseteq B$, $A \not\supseteq B$.

Функции

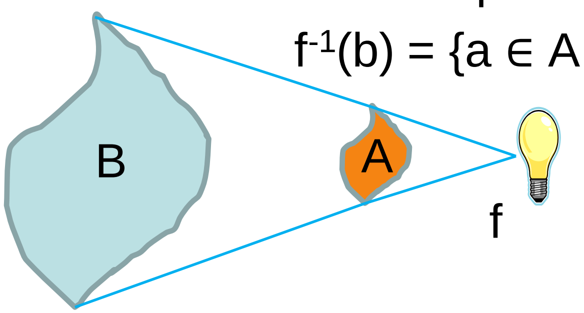
Определение Пусть A и B – два множества. Определим функцию f : A → B как отображение, которое каждому элементу a ∊ A ставит в соответствие элемент b ∊ B. Это записывается как b=f(a). Примем, что D(f) есть область определения функции f; R(f) – область определения значений функции; f(a) – область тех значений функции f, когда аргумент функции f пробегает множество A.

Замечание В данном определении функция f всюду определена. Частично определённая функция f : A → B есть отображение, которое каждому элементу из A сопоставляет не более одного элемента из множества B.

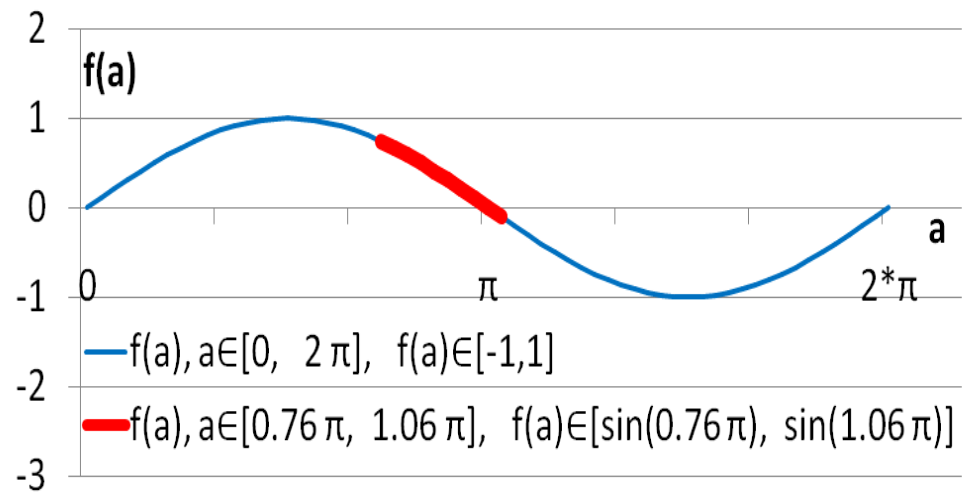
Отображение f : A → B

A есть прообраз для B:

$f^{-1}(b) = \{a \in A: f(a) = b\}$.



B есть образ для A: Im f = {f(a) : a ∊ A}.



Данное определение можно распространить на функции многих переменных f : A → B, при этом множество A будет является упорядоченным множеством (*кортежем*) множеств значений аргументов.

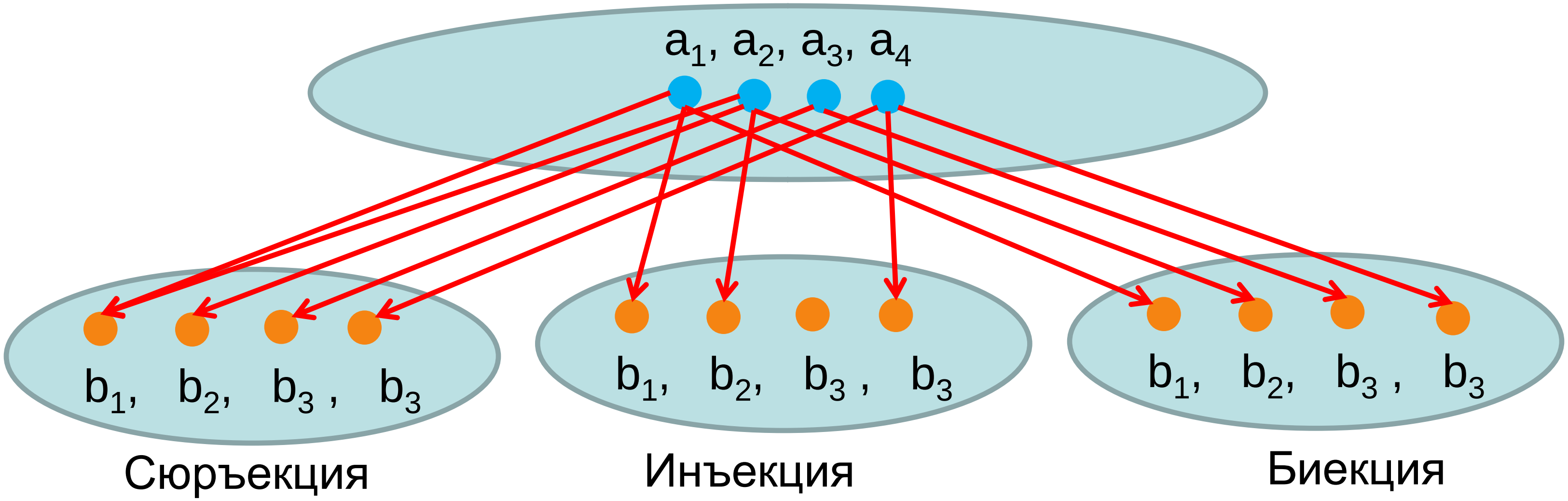
Сюръекция, инъекция, биекция

<u>*Определение*</u> Отображение f : A → B называют *сюръективным* если каждый элемент множества B является отображением хотя бы одного элемента из множества A: $\forall b \in B \to \exists a \in A \ (b = f(a))$.

<u>*Определение*</u> Отображение f : A → B называют *инъективным* если различные элементы образа B являются отображением различных элементов прообраза A: $\forall a_1, a_2 \in A \ ((a_1 \neq a_2) \to \ f(a1) \neq f(a2))$.

<u>*Определение*</u> Отображение являющееся одновременно *сюръективным* и *инъективным* есть *биективное* отображение.

$\forall b \ ((b \in B) \land (b = f(a_1)) \land (a_1 \in A)) \to \forall a_2 \ ((a_2 \in A) \land f(a_1) \neq f(a_2))$.



Для множеств $A = \{a_1, a_2, a_3, a_4\}$ и $B = \{b_1, \ b_2, \ b_3 , \ b_3\}$ показаны различные виды отображений.

Равенство, обратимость

<u>*Определение*</u> Функции $f : A \to B$ и $g : C \to D$ *равны*, если $A = C$ и $B = D$ и $f(a) = g(a)$, где $a \in A$.

<u>*Определение*</u> Функции $I_A : A \to A$, для которой $I(a) = a\ \forall a \in A$, называется *тождественной*.

<u>*Определение*</u> Если $f : A \to B$ и $C \subset A$, то функция $f : C \to B$, называется *сужением* функции f на множество C и обозначается $f|C$. Функция $f : A \to B$, называется *расширением* $f : C \to B$, где $C \subset A$.

<u>*Определение*</u> *Композиция* $g \quad f$ функций $f : A \to B$ и $g : B \to C$, есть функция

$f \quad g : A \to C$, для которой $(g \quad f)(a) = g(f(a))$, где $\forall a \in A$.

<u>*Пример*</u> Пусть $f(x) = y = x^2$, $g(y) = z = \sin(y)$, тогда композиция $g \quad f$, будет

$z = g(f(x)) = \sin(x^2)$.

<u>*Замечание*</u> Для тождественной функции $f \quad I_A = I_B \quad f = f$.

<u>*Определение*</u> Функция $f^{-1}: B \to A$ называется *обратной* к $f : A \to B$, если $f \quad f^{-1} = I_B$ и $f^{-1} \quad f = I_A$.

<u>*Утверждение*</u> Если обратная функция для функции f существует, то она единственная.

<u>*Теорема*</u> Функция $f : A \to B$ имеет обратную функцию тогда и только тогда когда отображение взаимно однозначно.

Упорядоченные множества

<u>*Определение*</u> Множества в которых порядок следования элементов имеет значение называют *упорядоченным множеством* или *кортежем*, перечисление элементов такого множества заключают в круглые () или угловые ⟨⟩ скобки, например: (a, b, c, d) или ⟨a, b, c, d⟩. Кортежи содержащие два элемента называют *упорядоченной двойкой*, три – *упорядоченной тройкой* и т. д.

<u>*Определение*</u> *Декартово произведение* двух множеств – это множество, элементами которого являются всевозможные кортежи (упорядоченные двойки) элементов исходных множеств.

Например декартово произведение A x B множеств

A = {a, b, c} и B = {0, 1}:

A x B = {(a, 0), (a, 1), (b, 0), (b, 1), (c, 0), (c, 1)}.

Эквивалентность множеств

<u>Определение</u> Множества А и В *эквивалентны* ($A \sim B$), если между их элементами возможно установить взаимно однозначное соответствие.

Свойства отношения эквивалентности

1. Рефлексивность $A \sim A$
2. Коммутативность $A \sim B \rightarrow B \sim A$
3. Транзитивность $(A \sim B) \wedge (B \sim C) \rightarrow (A \sim C)$

<u>Пример</u>

Множества $A = \{a, b, c, d\}$ и $B = \{00, 01, 10, 11\}$ эквивалентны $A \sim B$.

Мощность множества, счётные множества

<u>*Определение*</u> *Мощность множества* A (обозначается |A| или card(A)) есть класс эквивалентных ему множеств. Мощность конечного множества есть число его элементов.

<u>*Замечание*</u> Символ приписываемый классу эквивалентных множеств называют *кардинальным числом* или *кардиналом*.

<u>*Замечание*</u> Эквивалентные множества A и B равномощны, $A \sim B \leftrightarrow |B| = |A|$.

<u>*Определение*</u> Множество A *счётно*, если A эквивалентно множеству $\mathbb{N}$ натуральных чисел. В противном случае множество A *несчётно*.

<u>*Утверждение*</u> Из всякого безконечного множества можно выделить счётное множество.

<u>*Утверждение*</u> Объединение конечного или счётного множества счётных множеств счётно.

<u>*Замечание*</u> Мощности конечных множеств называют *финитными кардиналами*, мощности безконечных множеств называют *трансфинитными кардиналами*.

Множество A={a, b, c, d}, мощность множества |A| = 4.

Непрерывное множество или континуум

Утверждение Множество C всех последовательностей из 0 и 1 *несчётно*.

Определение Множество A имеет мощность *континуума* c, если A эквивалентно множеству всех безконечных последовательностей из 0 и 1.

Следствие Множество C всех безконечных последовательностей из 0 и 1 имеет мощность континуума: $|C| = c$.

Утверждение Если к безконечному множеству добавить конечное или счётное множество элементов, то его мощность не изменится.

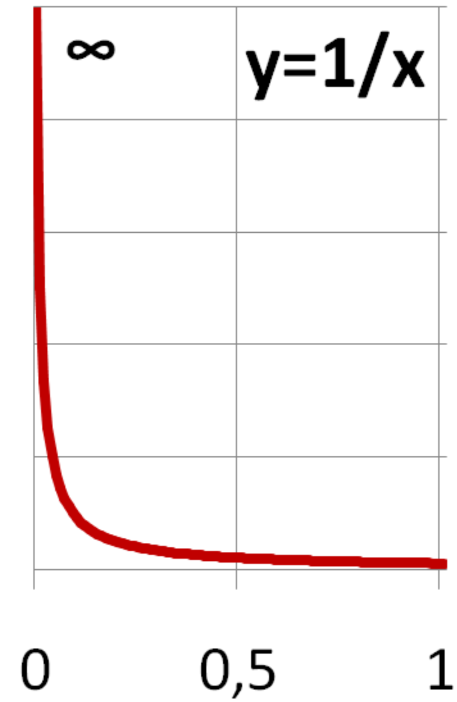
Доказательство Пусть A – безконечное множество, а B – конечное или счётное множество, причём $A \cap B = \varnothing$. Покажем, что $A \sim A \cup B$. Выделим из A счётное подмножество A_1, так, что $C = A - A_1$ тогда $A = C \cup A_1$, и $A \cup B = (C \cup A_1) \cup B = C \cup (A_1 \cup B)$. Так как $A_1 \cup B \sim A_1 \rightarrow A \cup B = C \cup (A_1 \cup B) \sim C \cup A_1 = A \rightarrow A \cup B \sim A$.

Утверждение Если A – несчётное множество, а B – конечное или счётное его подмножество, то $(A – B) \sim A$.

Доказательство Пусть $C = A – B$. Тогда $A = C \cup B$. Пусть B конечное или счётное, тогда для того, что бы A было несчётным, множество C должно быть несчётным. Следовательно $(A – B) \sim C$.

Мощность континуума

__Теорема__ Множество M = [a,b], где a<b, имеет мощность континуума с.



__Доказательство__ Доказательство теоремы вполне очевидно из рассмотрения функции вида f(x)=1/x на множестве X=(0,1], взаимно однозначно отображаемом на безконечное несчётное множество [1,∞).
Учитывая, что при помощи функции y = (b-a)·x+a возможно взаимно однозначно отобразить множество [a,b] на [0,1], можно показать, что любое множество M = [a,b], где a<b, имеет мощность континуума |[a,b]| = c.
__Замечание__ Множества |[a,b]|, |[a,b)|, |(a,b]|, |(a,b)|, где a<b равномощны и имеют мощность континуума:
|[a,b]| = |[a,b)| = |(a,b]| = |(a,b)| = c.

Можно найти достаточно функций отображающих ограниченный интервал на безконечный, например, класс функций имеющих вертикальные асимптоты (y=1/x, y=tg(x), y=ctg(x) и др.)

Сравнение мощностей

<u>*Определение*</u> Пусть A, B произвольные множества, а $|A|$ и $|B|$ их мощности, тогда возможны следующие сравнения мощностей:

$|A| = |B| \leftrightarrow A \sim B$

$|A| \leq |B| \leftrightarrow |B| \geq |A| \leftrightarrow A \sim C$, где C некоторое подмножество B, $C \subseteq B$

$|A| < |B| \leftrightarrow |B| > |A| \leftrightarrow A \sim C$, где C некоторое подмножество B, $C \subset B$

<u>*Замечание*</u> Случай, когда множество A не эквивалентно никакому подмножеству множества B, а множество B не эквивалентно никакому подмножеству A, невозможен.

<u>*Теорема*</u> Если множество $A \sim B_1$ ($B_1 \subseteq B$), а множество $B \sim A_1$ ($A_1 \subseteq A$), то множества A и B эквивалентны ($A \sim B$) или кратко: $|A| \leq |B| \wedge |B| \leq |A| \leftrightarrow |A| = |B|$.

Шкала мощностей

Теорема Пусть A некоторое множество и P(A) есть множество всех подмножеств множества A, тогда $|A| \leq |P(A)|$.

Отношение $|A| \leq |P(A)|$ следует из того, что если каждому элементу $a \in A$ сопоставить одноэлементные подмножества $\{a\} \in P(A)$ то уже будет достигнут порог равномощности, добавление последующих подмножеств из P(A) приведёт превышению мощность P(A) по отношению к A. Для непустых множеств $A \neq \varnothing$ (напомним, что пустое множество является подмножеством всех множеств), возможно записать:

Теорема (Кантора) $|A| < |P(A)|$.

Доказательство теоремы следует из вышеприведённых рассуждений о следует из достижения порога равномощности, если каждому элементу $a \in A$ сопоставить одноэлементные подмножества $\{a\} \in P(A)$.

Множество P(A) всех подмножеств A можно обозначить через 2^A, тогда мощность $|P(A)| = 2^{|A|}$, в соответствии с теоремой $|A| < 2^{|A|}$.

Таким образом, беря за основу произвольное множество A возможно сформировать последовательность кардинальных чисел: $|A| < 2^{|A|} < 2^{2\wedge|A|} < \ldots$

Выбирая в качестве основы множество $\mathbb{N}$ натуральных чисел, можно построить последовательность континуумных кардиналов, для счётных множеств:

$|\mathbb{N}| = c < 2^c < 2^{2\wedge c} < \ldots$, называемых соответственно континуум, гиперконтинуум, гипер-гиперконтинуум и т.д.

Отношения эквивалентности

Пусть $A_1, A_2, \ldots, A_n$ – произвольные множества, вообще говоря, разнородные.

<u>*Определение*</u> *n-арное отношение* ρ^n на множествах $A_1, A_2, \ldots, A_n$ есть подмножество декартова произведения $A_1 \times A_2 \times \ldots \times A_n$.

<u>*Определение*</u> Отношение конечно, если оно состоит из конечного числа элементов.

<u>*Определение*</u> Пусть A – произвольное множество, тогда бинарное отношение

$s \subseteq A \times A$ есть *отношение эквивалентности* (обозначение $a \sim b$), если оно удовлетворяет следующим аксиомам $\forall a, b, c \in A$:

1. $a \sim a$, рефлексивность;
2. $a \sim b \rightarrow b \sim a$, коммутативность;
3. $a \sim b \wedge b \sim c \rightarrow a \sim c$, транзитивность.

<u>*Определение*</u> *Разбиение множества* A есть семейство непустых попарно непересекающихся подмножеств $A_1, A_2, \ldots, A_n$ из A, в объединении дающих A. Подмножества $A_1, A_2, \ldots, A_n$ называются смежными классами разбиения.

<u>*Пример*</u> $A = \{a,b,c,d\} = \{a,b\} \cup \{c\} \cup \{d\}$.

<u>*Теорема*</u> Каждому отношению эквивалентности, определённому на множестве A, соответствует некоторое разбиение множества A. Каждому разбиению множества A соответствует некоторое отношение эквивалентности. Или можно сказать, что между классом всех определённых на множестве A эквивалентностей и классом всех разбиений множества A существует взаимно однозначное соответствие.

Отношения эквивалентности

Замечания

1. Разбиение множества А на одноэлементные подмножества и разбиение А, состоящие из одного только множества А, называют тривиальными (несобственными) разбиениями.

2. Разбиение А на одноэлементные подмножества соответствует отношению эквивалентности, которое есть равенство.

3. Разбиение множества А, состоящее из одного только множества А, соответствует отношению эквивалентности, содержащему всё множество А x А.

4. $a \sim b \leftrightarrow M_a = M_b$, где M_a множество всех элементов $x \sim a$, M_b множество всех элементов $y \sim b$.

Определение Совокупность классов эквивалентности множества А называется *фактор-множеством* A/s множества А по эквивалентности s.

Определение Отображение $p : A \to A/s$, при котором $p(a) = M_a$, называется *каноническим* (естественным).

Парадокс Рассела

<u>*Формулировка*</u> Пусть К – *множество всех множеств*, *которые не содержат себя в качестве своего элемента*. Содержит ли К само себя в качестве элемента? Если предположить, что содержит, то мы получаем противоречие с «*не содержат себя в качестве своего элемента»*. Если предположить, что К не содержит себя как элемент, то вновь возникает противоречие, ведь К – множество всех множеств, которые не содержат себя в качестве своего элемента, а значит должно содержать все возможные элементы, *включая и себя*.

<u>*Парадокс брадобрея*</u>

Те, которые не бреются сами, бреются у брадобрея. А как быть с брадобреем?

<u>*Парадокс лжеца*</u>

Некто сказал, что он всегда лжёт. Солгал ли он в этот раз?

Каноническое разложение функции

Пусть $f : A \to B$ есть некоторая функция. Определим на A отношение $s \subseteq A \times A$, положив $\forall a,a' \in A$ $(a \sim a' \leftrightarrow f(a) = f(a'))$. Отношение s есть отношение эквивалентности, т.к. $\forall a, b, c \in A$:

1. $f(a) = f(a) \to a \sim a$
2. $(f(a) = f(b) \to f(b) = f(a)) \to (a \sim b \to b \sim a)$
3. $(f(a) = f(b) \wedge f(b) = f(c) \to f(a) = f(c)) \to (a \sim b \wedge b \sim c \to a \sim c)$

Введённое отношение s называют *ядерной эквивалентностью* для отображения f. Классы эквивалентности A/s есть полные прообразы элементов множества B при отображении f, то есть: $A_b = f^{-1}(b) = \{a \in A: f(a)=b\}$.

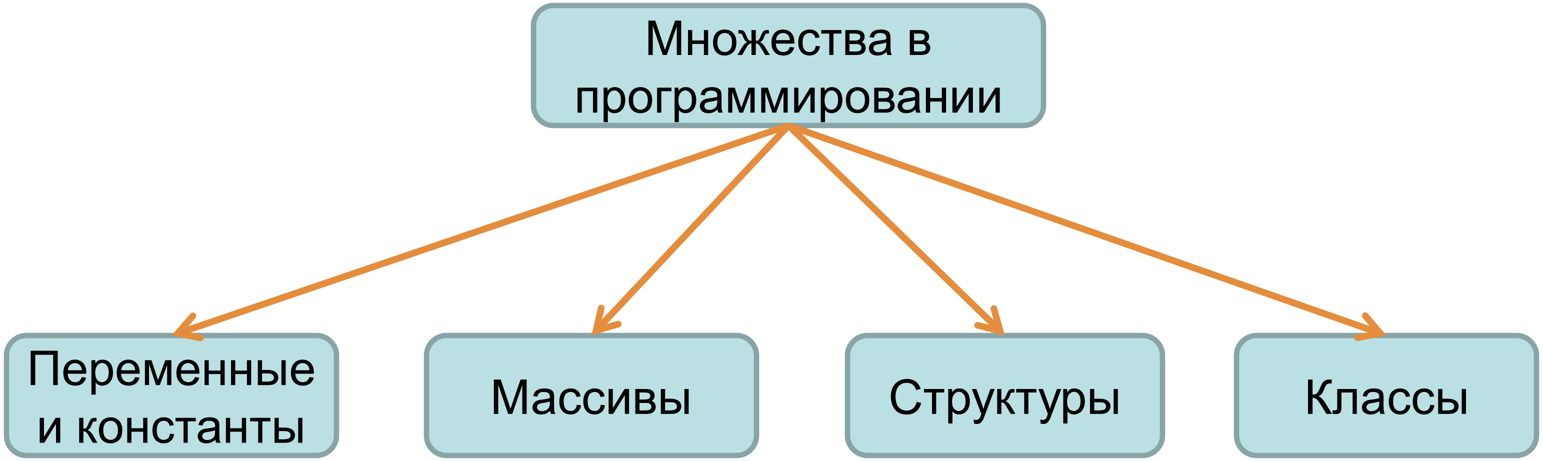
Отображение f можно разложить в композицию двух отображений: $f = q \quad p$, т.е.

$f(a) = q(p(a))$, представление $f = q \quad p$ называется *каноническим разложением* функции f.

Пример канонического разложения функции

Пример Получить каноническое разложение функции $f : A \to B$,
$f = \{(0,a),(1,b),(2,c),(3,b),(4,a),(5,d),(6,b),(7,a),(8,e)\}$,
область определения функции $D(f) = A = \{0, 1, 2, 3, 4, 5, 6, 7, 8\}$,
область значений $Im(f) = B = \{a, b, c, d, e\}$, тогда классы эквивалентности:
$M_a = [a]_s = f^{-1}(a) = \{0, 4, 7\}$, $q(M_a) = a$
$M_b = [b]_s = f^{-1}(b) = \{1, 3, 6\}$, $q(M_b) = b$
$M_c = [c]_s = f^{-1}(c) = \{2\}$, $q(M_c) = c$
$M_d = [d]_s = f^{-1}(d) = \{5\}$, $q(M_d) = d$
$M_e = [e]_s = f^{-1}(e) = \{8\}$, $q(M_e) = e$
Зададим функции p и q:
$p(A) = M_{f(A)} = \{(0,M_a),(1,M_b),(2,M_c),(3,M_b),(4,M_a),(5,M_d),(6,M_b),(7,M_a),(8,M_e)\}$,
$D(p) = A$, $Im(p) = \{M_a, M_b, M_c, M_d, M_e\}$
$q(K_B) = \{(M_a, a), (M_b, b), (M_c, c), (M_d,d), (M_e, e)\}$
$f(x) = q(p(x)), \forall x \in A$

Приложения множеств в программировании



Одномерный массив, выделение и очистка памяти C++

```
int main() {// функция входа в программу на C++
  // объявление указателя и переменных
  char* arr1d;              // указатель
  unsigned int Xsize = 16; // размер массива
  // выделение памяти под массив
  arr1d = new char[Xsize];
  // инициализация нулём
  memset(arr1d, 0, Xsize * sizeof(char));

  //...
  // какие-то действия

  // чистим память
  if( arr1d ) {
    // если указатель не нулевой
    delete[]arr1d;// удаляем массив
    arr1d = 0;    // обнуляем указатель
  }
  // выход
  return 0;
}
```

адрес (пример)	индекс	значение
0x009ECDB8	0	0
0x009ECDB9	1	0
0x009ECDBA	2	0
0x009ECDBB	3	0
0x009ECDBC	4	0
0x009ECDBD	5	0
0x009ECDBE	6	0
0x009ECDBF	7	0
0x009ECDC0	8	0
0x009ECDC1	9	0
0x009ECDC2	10	0
0x009ECDC3	11	0
0x009ECDC4	12	0
0x009ECDC5	13	0
0x009ECDC6	14	0
0x009ECDC7	15	0

В C++имя массива является указателем хранящим адрес нулевого (начального) элемента массива. Индексы массива соответствуют смещению в логическом адресном пространстве от начала массива.

Двумерный массив, выделение и очистка памяти C++

```
int main() {    // функция входа в программу на C++
  // необходимые указатели и переменные
  char** arr2d;// объявление указателя указателей
              //  на массивы типа char
  unsigned int Xsize = 32; // размеры массива (X)
  unsigned int Ysize = 32; // размеры массива (Y)
  // выделение памяти для массива массивов
  arr2d = new char*[Ysize]; // для указателей
  for (unsigned int i = 0; i < Ysize; i++) {
    arr2d[i] = new char[Xsize]; // для данных
    // инициализация памяти нулём
    memset(arr2d[i], 0, Xsize * sizeof(char));
  }
  // ...
  // какие-то действия
  // обращение к элементам двумерного массива
  // может быть такое:
  //  arr2d[y][x], y=0...Ysize-1, x=0...Xsize-1
  // ...
  // чистим память
  if( arr2d ) { // если указатель не нулевой
    for (unsigned int i = 0; i < Ysize; i++)
      delete[]arr2d[i]; // массивы данных
    delete[]arr2d;      // массив указателей
    arr2d = 0;          // обнуляем указатель
  }
  return 0;     // выход
}
```

массив указателей		массивы данных					
индекс	**значение**						
0	adr0	индекс	0	1	. . .	Xsize-2	Xsize-1
		значение	x	x	. . .	x	x
1	adr1	индекс	0	1	. . .	Xsize-2	Xsize-1
		значение	x	x	. . .	x	x
. . .	. . .	. . .			. . .		. . .
Ysize-2	adr(Y-2)	индекс	0	1	. . .	Xsize-2	Xsize-1
		значение	x	x	. . .	x	x
Ysize-1	adr(Y-1)	индекс	0	1	. . .	Xsize-2	Xsize-1
		значение	x	x	. . .	x	x

С точки зрения разработчика на C++, с учётом линейной модели памяти ЭВМ, двумерный массив является массивом указателей указывающих на массивы данных, т.е. массивом массивов.

Двумерный массив является примером упорядоченного множества множеств (кортежем множеств, точнее кортежем кортежей):

$M = (A_0, A_1, A_2, \ldots, A_n)$, где $A_0=(a_{00},a_{01}, \ldots, a_{0m})$, …, $A_n=(a_{n0},a_{n1}, \ldots, a_{nm})$.

Ассоциативный массив

```
#include <iostream>
#include <string>
#include <map>
// объявление типа структуры для данных о планете
typedef struct  {
  unsigned int numb;  // номер
  float weight;       // масса, кг
  float radius;       // средний радиус, км
} plan_data;
// объявление типа ассоциативного контейнера map
// отображающего тип std::string на структуру plan_data
typedef std::map <std::string, plan_data> MapPlaet;
//---------------------------------------------------
int main( ) {
  MapPlaet cont; // объявление экземпляра типа MapPlaet
  // присвоение значений элементам контейнера maparr
  cont["Mercury"].numb   = 1;
  cont["Mercury"].weight = 3.33022E23;
  cont["Mercury"].radius = 2439.7;
  cont["Venus"].numb     = 2;
  cont["Venus"].weight   = 4.8685E24;
  cont["Venus"].radius   = 6051.8;
  cont["Earth"].numb     = 3;
  cont["Earth"].weight   = 5.9736E24;
  cont["Earth"].radius   = 6371.0;
  // вывод данных на экран, перебор через итератор
  for( MapPlaet::iterator pm = cont.begin(); pm != cont.end(); pm++) {
    std::cout<<"Planet name: "<<(*pm).first<<std::endl;
    std::cout<<"\tnumber = "<<(*pm).second.numb<<std::endl;
    std::cout<<"\tweight, kg = "<<(*pm).second.weight<<std::endl;
    std::cout<<"\taverage radius, km = "<<(*pm).second.radius<<std::endl;
  }
  cont.clear();  // очистка контейнера cont
  return 0;      // выход
}
```

Вывод программы

```
Planet name: Earth
          number = 3
          weight, kg = 5.9736e+24
          average radius, km = 6371
Planet name: Mercury
          number = 1
          weight, kg = 3.33022e+23
          average radius, km = 2439.7
Planet name: Venus
          number = 2
          weight, kg = 4.8685e+24
          average radius, km = 6051.8
```

$N \sim D \rightarrow |N| = |D|$

$N = (n_0, n_1, \dots , n_{m-1})$

$n_{i\ (i=0\dots m-1)}$: строка

$D = (D_0, D_1, \dots D_{m-1})$

$D_{i\ (i=0\dots m-1)} = (a, b, c)$

a: целое беззнакоковое

b: с плавающей точкой

c: с плавающей точкой

Список литературы

1. Набебин А. А. Дискретная математика. – М.: Научный мир, 2010. 512 с.: ил.
2. Андерсон Дж. А. Дискретная математика и комбинаторика. : Пер с англ. – М. Издательский дом «Вильямс», 2004. – 960 с.: ил. – Паралл. тит. англ.
3. Фомичёв В. М. Методы дискретной математики в криптологии. – М.: Диалог-МИФИ, 2010. – 424 с.: ил.
4. Прата С. Язык программирования С++. Лекции и упражнения, 6-е изд.: Пер. с англ. – М.: ООО «И.Д. Вильямс», 2013. – 1248 с.: ил. – Парал. тит. англ.
5. Либерти Д., Бредли Д. Освой самостоятельно С++ за 21 день, 5-е издание. : Пер. с англ. – М.: Издательский дом «Вильямс», 2012. – 768 с.:ил.
6. Лафорте Р. Объектно-ориентированное программирование в С++. Класика Computer Science. 4-е изд. – СПб.: Питер, 2012. – 928 с.: ил.

Суриков В. И. «Переход Суворова через Альпы»
Василий Иванович Суриков
(12 (24) января 1848, Красноярск — 6 (19) марта 1916, Москва) — русский живописец, мастер масштабных исторических полотен.