Математическая логика

Конструкции над множествами

Валерий Исаев

29 февраля 2016 г.

План лекции

Теорема Кантора-Бернштейна

Separation Axiom

Фактор-множества

Зависимые множества

Формулировка и доказательство

Theorem

Если множество A вкладывается в множество B, и наоборот, то A и B равномощны.
Другими словами, если $|A| \leq |B|$ и $|B| \leq |A|$, то $|A| = |B|$.

Формулировка и доказательство

Доказательство.

Пусть $f : A \to B$ и $g : B \to A$ – вложения. Тогда определим $h : A \to A$ как $h = g \circ f$. Пусть $A_0 = A$, $A_1 = g(B)$, $A_{k+2} = h(A_k)$. Так как $|A_1| = |B|$, нам достаточно показать, что $|A_0| = |A_1|$. Пусть $C_i = A_i \setminus A_{i+1}$. Если ограничить функцию h на C_i, то мы получим биекцию $h_i : C_i \to C_{i+2}$. Теперь можно определить биекцию $t : A_0 \to A_1$ как $t(x) = h_i(x)$, если $x \in C_i$ и i – четно, иначе $t(x) = x$. □

Алгоритм Кантора-Бернштейна

- Является ли данное доказательство конструктивным?

Алгоритм Кантора-Бернштейна

- Является ли данное доказательство конструктивным?
- Нет, но если мы добавим несколько условий, то да.

Алгоритм Кантора-Бернштейна

- Является ли данное доказательство конструктивным?
- Нет, но если мы добавим несколько условий, то да.
- Во-первых, предположим, что образ A в B разрешим, и наоборот.

Алгоритм Кантора-Бернштейна

- Является ли данное доказательство конструктивным?
- Нет, но если мы добавим несколько условий, то да.
- Во-первых, предположим, что образ A в B разрешим, и наоборот.
- Во-вторых, предположим, что пересечение всех A_i разрешимо в A.

Алгоритм Кантора-Бернштейна

- Является ли данное доказательство конструктивным?
- Нет, но если мы добавим несколько условий, то да.
- Во-первых, предположим, что образ A в B разрешим, и наоборот.
- Во-вторых, предположим, что пересечение всех A_i разрешимо в A.
- Тогда эта теорема доказуема без исключенного третьего.

Примеры

- Множества $\mathbb{N}$ и $\mathbb{N} \times \mathbb{N}$ равномощны.

Примеры

- Множества $\mathbb{N}$ и $\mathbb{N} \times \mathbb{N}$ равномощны.
- Мы можем вложить $\mathbb{N}$ в $\mathbb{N} \times \mathbb{N}$ как $x \mapsto (x, 0)$.

Примеры

- Множества $\mathbb{N}$ и $\mathbb{N} \times \mathbb{N}$ равномощны.
- Мы можем вложить $\mathbb{N}$ в $\mathbb{N} \times \mathbb{N}$ как $x \mapsto (x, 0)$.
- И наоборот, $(x, y) \mapsto 2^x \cdot 3^y$.

Примеры

- Множества $\mathbb{N}$ и $\mathbb{N} \times \mathbb{N}$ равномощны.
- Мы можем вложить $\mathbb{N}$ в $\mathbb{N} \times \mathbb{N}$ как $x \mapsto (x, 0)$.
- И наоборот, $(x, y) \mapsto 2^x \cdot 3^y$.
- Соответствующие подмножества $\mathbb{N}$ и $\mathbb{N} \times \mathbb{N}$ разрешимы, и пересечение всех A_i пусто. Следовательно, биекция вычислима.

План лекции

Теорема Кантора-Бернштейна

Separation Axiom

Фактор-множества

Зависимые множества

Separation axiom для множеств

- Пусть A – множество и $P(a)$ – предикат на нем.

Separation axiom для множеств

- Пусть A – множество и $P(a)$ – предикат на нем.
- Тогда существует множество $\{a \in A \mid P(a)\}$.

Separation axiom для множеств

- Пусть A – множество и $P(a)$ – предикат на нем.
- Тогда существует множество $\{a \in A \mid P(a)\}$.
- Это множество является подмножеством A и состоит в точности из тех его элементов, которые удовлетворяют предикату P.

Separation axiom для множеств

- Пусть A – множество и $P(a)$ – предикат на нем.
- Тогда существует множество $\{a \in A \mid P(a)\}$.
- Это множество является подмножеством A и состоит в точности из тех его элементов, которые удовлетворяют предикату P.
- Пример: множество четных чисел можно определить как

$$Even = \{n \in \mathbb{N} \mid \exists k \in \mathbb{N}\ (2k = n)\}$$

Separation axiom для множеств

- Пусть A – множество и $P(a)$ – предикат на нем.
- Тогда существует множество $\{a \in A \mid P(a)\}$.
- Это множество является подмножеством A и состоит в точности из тех его элементов, которые удовлетворяют предикату P.
- Пример: множество четных чисел можно определить как

$$Even = \{n \in \mathbb{N} \mid \exists k \in \mathbb{N}\ (2k = n)\}$$

- Пример: если $f : A \to B$, то *образ* f – это подмножество B, которое определяется как

$$im(f) = \{b \in B \mid \exists a \in A\ (f(a) = b)\}.$$

Separation axiom в программировании

- В большинстве языков программирования нет аналогичной конструкции.

Separation axiom в программировании

- В большинстве языков программирования нет аналогичной конструкции.
- Очень жалко их таких, ибо полезная конструкция.

Separation axiom в программировании

- В большинстве языков программирования нет аналогичной конструкции.
- Очень жалко их таких, ибо полезная конструкция.
- Часто функции определены не для всех значений типа, а только на некотором подмножестве. Если передавать элемент не из этого подмножества, то функция либо выбросит исключение, либо поведение вообще будет неопределено.

Separation axiom в программировании

- В большинстве языков программирования нет аналогичной конструкции.
- Очень жалко их таких, ибо полезная конструкция.
- Часто функции определены не для всех значений типа, а только на некотором подмножестве. Если передавать элемент не из этого подмножества, то функция либо выбросит исключение, либо поведение вообще будет неопределено.
- Например, очень было бы полезно в java/C#/... иметь тип $\{x \in A \mid x \neq \mathbf{null}\}$.

Separation axiom в программировании

- В большинстве языков программирования нет аналогичной конструкции.
- Очень жалко их таких, ибо полезная конструкция.
- Часто функции определены не для всех значений типа, а только на некотором подмножестве. Если передавать элемент не из этого подмножества, то функция либо выбросит исключение, либо поведение вообще будет неопределено.
- Например, очень было бы полезно в java/C#/... иметь тип $\{x \in A \mid x \neq \mathbf{null}\}$.
- Другой пример: типы вида $\{x \in \mathbf{int} \mid a < x < b\}$.

План лекции

Мотивация

- Часто различные элементы некоторого множества должны быть одинаковыми.

Мотивация

- Часто различные элементы некоторого множества должны быть одинаковыми.
- Например, рациональное число можно задать как пару целых чисел (числитель и знаменатель), но функция $\mathbb{Z} \times \mathbb{Z}_{>0} \to \mathbb{Q}$ не является биекцией, так как некоторые пары целых чисел задают одно и то же рациональное число.

Мотивация

- Часто различные элементы некоторого множества должны быть одинаковыми.
- Например, рациональное число можно задать как пару целых чисел (числитель и знаменатель), но функция $\mathbb{Z} \times \mathbb{Z}_{>0} \to \mathbb{Q}$ не является биекцией, так как некоторые пары целых чисел задают одно и то же рациональное число.
- Конструкция фактор множеств позволяет объединить различные элементы в один.

Характеризация

- Пусть на множестве A есть отношение эквивалентности $\sim$.

Характеризация

- Пусть на множестве A есть отношение эквивалентности $\sim$.
- Тогда для любого $a \in A$ множество $A/{\sim}$ должно содержать элемент $[a]_{\sim}$.

Характеризация

- Пусть на множестве A есть отношение эквивалентности $\sim$.
- Тогда для любого $a \in A$ множество $A/{\sim}$ должно содержать элемент $[a]_\sim$.
- Более того, должно быть верно, что $[a]_\sim = [a']_\sim$ тогда и только тогда, когда $a \sim a'$.

Характеризация

- Пусть на множестве A есть отношение эквивалентности $\sim$.
- Тогда для любого $a \in A$ множество $A/\sim$ должно содержать элемент $[a]_\sim$.
- Более того, должно быть верно, что $[a]_\sim = [a']_\sim$ тогда и только тогда, когда $a \sim a'$.
- Мы могли бы добавить аксиому, которая говорит, что это множество существует, но этого не нужно делать, так как его существование следует из других аксиом.

Определение

- Пусть $a \in A$. Тогда *класс эквивалентности* a – это подмножество A, состоящее из таких a', что $a \sim a'$.

Определение

- Пусть $a \in A$. Тогда *класс эквивалентности* a – это подмножество A, состоящее из таких a', что $a \sim a'$.
- Класс эквивалентности a обозначается $[a]_\sim$. Его определение можно записать в виде формулы:

$$[a]_\sim = \{a' \in A \mid a \sim a'\}$$

Определение

- Пусть $a \in A$. Тогда *класс эквивалентности* a – это подмножество A, состоящее из таких a', что $a \sim a'$.
- Класс эквивалентности a обозначается $[a]_\sim$. Его определение можно записать в виде формулы:

$$[a]_\sim = \{a' \in A \mid a \sim a'\}$$

- Теперь мы можем определить $A/\sim$ как

$$A/\sim\, = \{C \in \mathcal{P}(A) \mid \exists a \in A\ (C = [a]_\sim)\}$$

Рациональные числа

- Мы можем определить отношение эквивалентности на множестве $\mathbb{Z} \times \mathbb{Z}_{>0}$ как

$$(x, y) \sim (x', y') \Leftrightarrow x \cdot y' = x' \cdot y$$

Рациональные числа

- Мы можем определить отношение эквивалентности на множестве $\mathbb{Z} \times \mathbb{Z}_{>0}$ как

$$(x, y) \sim (x', y') \Leftrightarrow x \cdot y' = x' \cdot y$$

- Теперь множество рациональных чисел можно определить как $\mathbb{Q} = \mathbb{Z} \times \mathbb{Z}_{>0} / \sim$.

Функции над фактор-множествами

- Обратите внимание, что некоторые “функции” над множеством вида $A/\sim$ не являются корректными функциями.

Функции над фактор-множествами

- Обратите внимание, что некоторые "функции" над множеством вида $A/\sim$ не являются корректными функциями.
- Например, "функция" $nom : \mathbb{Q} \to \mathbb{Z}$, $nom([(x, y)]_\sim) = x$, возвращающая числитель, не является функцией.

Функции над фактор-множествами

- Обратите внимание, что некоторые "функции" над множеством вида $A/\sim$ не являются корректными функциями.
- Например, "функция" $nom : \mathbb{Q} \to \mathbb{Z}$, $nom([(x,y)]_\sim) = x$, возвращающая числитель, не является функцией.
- Проблема в том, что она не сохраняет равенство. Если бы она была корректной функцией, то $1 = 2$:

$$1 = nom(1/2) = nom(2/4) = 2$$

Корректность функций над фактор-множествами

- Мы можем определить эту функцию, если немного изменим ее описание: $nom([(x, y)]_{\sim}) = x/gcd(x, y)$.

Корректность функций над фактор-множествами

- Мы можем определить эту функцию, если немного изменим ее описание: $nom([(x, y)]_\sim) = x/gcd(x, y)$.
- Чтобы проверить, что это определение является корректным, нам нужно проверить, что функция сохраняет эквивалентность. То есть если $(x, y) \sim (x', y')$, то $nom([(x, y)]_\sim) = nom([(x', y')]_\sim)$.

Корректность функций над фактор-множествами

- Мы можем определить эту функцию, если немного изменим ее описание: $nom([(x, y)]_\sim) = x/gcd(x, y)$.
- Чтобы проверить, что это определение является корректным, нам нужно проверить, что функция сохраняет эквивалентность. То есть если $(x, y) \sim (x', y')$, то $nom([(x, y)]_\sim) = nom([(x', y')]_\sim)$.
- Действительно, чтобы доказать это, достаточно разложить все числа в произведения простых.

Фактор-типы в программировании

- Как и в случае с separation аксиомой, в большинстве языков прорамммирования нет конструкции фактор-типов.

Фактор-типы в программировании

- Как и в случае с separation аксиомой, в большинстве языков прораммирования нет конструкции фактор-типов.
- Но их можно смоделировать в некотором роде.

Фактор-типы в программировании

- Как и в случае с separation аксиомой, в большинстве языков прораммирования нет конструкции фактор-типов.
- Но их можно смоделировать в некотором роде.
- Конкретно, мы обычно переопределяем операцию сравнения на типе.

Канонические представители

- Иногда фактор-множества можно определить другим способом, используя понятие *канонического представителя*.

Канонические представители

- Иногда фактор-множества можно определить другим способом, используя понятие *канонического представителя*.
- Во-первых, для любого класса эквивалентности $[a]_\sim$ выбираем некоторый элемент из него $a_0 \in [a]_\sim$, который мы будем называть *каноническим представителем* этого класса.

Канонические представители

- Иногда фактор-множества можно определить другим способом, используя понятие *канонического представителя*.
- Во-первых, для любого класса эквивалентности $[a]_\sim$ выбираем некоторый элемент из него $a_0 \in [a]_\sim$, который мы будем называть *каноническим представителем* этого класса.
- Теперь мы можем определить $A/\sim$ как подмножество A, состоящее из канонических представителей.

Канонические представители

- Иногда фактор-множества можно определить другим способом, используя понятие *канонического представителя*.
- Во-первых, для любого класса эквивалентности $[a]_\sim$ выбираем некоторый элемент из него $a_0 \in [a]_\sim$, который мы будем называть *каноническим представителем* этого класса.
- Теперь мы можем определить $A/\sim$ как подмножество A, состоящее из канонических представителей.
- В случае с рациональными числами, канонический представитель $[(x, y)]_\sim$ – это $(x/gcd(x, y), y/gcd(x, y))$.

План лекции

Теорема Кантора-Бернштейна

Separation Axiom

Фактор-множества

Зависимые множества

Зависимые суммы

- Еще одна аксиома теории множеств говорит о том, что существуют (бесконечные) объединения множеств.

Зависимые суммы

- Еще одна аксиома теории множеств говорит о том, что существуют (бесконечные) объединения множеств.
- Вместо обычных конечных объединений мы добавили размеченные объединения.

Зависимые суммы

- Еще одна аксиома теории множеств говорит о том, что существуют (бесконечные) объединения множеств.
- Вместо обычных конечных объединений мы добавили размеченные объединения.
- По этой же причине вместо бесконечных объединений мы добавим бесконечные размеченные объединения, которые называются обычно *зависимой суммой* множеств.

Определение

- Размеченные объединения отличаются от обычных тем, что мы добавляем к ккаждому элементу метку, которая говорит нам из какого множества пришел этот элемент.

Определение

- Размеченные объединения отличаются от обычных тем, что мы добавляем к ккаждому элементу метку, которая говорит нам из какого множества пришел этот элемент.
- То есть, элементы $B_0 \amalg B_1$ можно задавать как $(0, b)$ и $(1, b')$, где $b \in B_0$ и $b' \in B_1$.

Определение

- Размеченные объединения отличаются от обычных тем, что мы добавляем к ккаждому элементу метку, которая говорит нам из какого множества пришел этот элемент.
- То есть, элементы $B_0 \amalg B_1$ можно задавать как $(0, b)$ и $(1, b')$, где $b \in B_0$ и $b' \in B_1$.
- Точно так же, элементы $B_0 \amalg \ldots \amalg B_n$ можно описать как (i, b_i), где $0 \leq i \leq n$ и $b_i \in B_i$.

Определение

- Размеченные объединения отличаются от обычных тем, что мы добавляем к ккаждому элементу метку, которая говорит нам из какого множества пришел этот элемент.
- То есть, элементы $B_0 \amalg B_1$ можно задавать как $(0, b)$ и $(1, b')$, где $b \in B_0$ и $b' \in B_1$.
- Точно так же, элементы $B_0 \amalg \ldots \amalg B_n$ можно описать как (i, b_i), где $0 \leq i \leq n$ и $b_i \in B_i$.
- Если у нас есть бесконечная последовательность множеств B_0, B_1, ..., то можно определить множество $B_0 \amalg B_1 \amalg \ldots$ как множество пар вида (i, b_i), где $i \in \mathbb{N}$, $b_i \in B_i$.

Определение

- Размеченные объединения отличаются от обычных тем, что мы добавляем к ккаждому элементу метку, которая говорит нам из какого множества пришел этот элемент.
- То есть, элементы $B_0 \amalg B_1$ можно задавать как $(0, b)$ и $(1, b')$, где $b \in B_0$ и $b' \in B_1$.
- Точно так же, элементы $B_0 \amalg \ldots \amalg B_n$ можно описать как (i, b_i), где $0 \leq i \leq n$ и $b_i \in B_i$.
- Если у нас есть бесконечная последовательность множеств B_0, B_1, ..., то можно определить множество $B_0 \amalg B_1 \amalg \ldots$ как множество пар вида (i, b_i), где $i \in \mathbb{N}$, $b_i \in B_i$.
- В общем случае, если у нас есть коллекция множеств B_a, зависящая от элементов $a \in A$, то мы можем определить множество $\Sigma(a \in A) B_a$ как множество пар (a, b), где $a \in A$ и $b \in B_a$.

Примеры

- Если B_a не зависит от A, то есть $B_a = B$ для всех $a \in A$, то $\Sigma(a \in A)B_a = A \times B$.

Примеры

- Если B_a не зависит от A, то есть $B_a = B$ для всех $a \in A$, то $\Sigma(a \in A)B_a = A \times B$.
- Если A – конечно, и для любого $a \in A$, B_a – тоже конечно, то $\Sigma(a \in A)B_a$ тоже будет конечно, и $|\Sigma(a \in A)B_a| = \sum\limits_{a \in A} |B_a|$.

Примеры

- Если B_a не зависит от A, то есть $B_a = B$ для всех $a \in A$, то $\Sigma(a \in A)B_a = A \times B$.
- Если A – конечно, и для любого $a \in A$, B_a – тоже конечно, то $\Sigma(a \in A)B_a$ тоже будет конечно, и $|\Sigma(a \in A)B_a| = \sum\limits_{a \in A} |B_a|$.
- В программировании такой тип тоже был бы полезен.

Примеры

- Если B_a не зависит от A, то есть $B_a = B$ для всех $a \in A$, то $\Sigma(a \in A)B_a = A \times B$.
- Если A – конечно, и для любого $a \in A$, B_a – тоже конечно, то $\Sigma(a \in A)B_a$ тоже будет конечно, и $|\Sigma(a \in A)B_a| = \sum_{a \in A} |B_a|$.
- В программировании такой тип тоже был бы полезен.
- Пусть $List(A)$ – тип список с элементами типа A.

Примеры

- Если B_a не зависит от A, то есть $B_a = B$ для всех $a \in A$, то $\Sigma(a \in A)B_a = A \times B$.
- Если A – конечно, и для любого $a \in A$, B_a – тоже конечно, то $\Sigma(a \in A)B_a$ тоже будет конечно, и $|\Sigma(a \in A)B_a| = \sum\limits_{a \in A} |B_a|$.
- В программировании такой тип тоже был бы полезен.
- Пусть $List(A)$ – тип список с элементами типа A.
- Пусть $Vec(A, n) = \{xs \in List(A) \mid length(xs) = n\}$ – тип списков длинны n.

Примеры

- Если B_a не зависит от A, то есть $B_a = B$ для всех $a \in A$, то $\Sigma(a \in A)B_a = A \times B$.
- Если A – конечно, и для любого $a \in A$, B_a – тоже конечно, то $\Sigma(a \in A)B_a$ тоже будет конечно, и $|\Sigma(a \in A)B_a| = \sum\limits_{a \in A} |B_a|$.
- В программировании такой тип тоже был бы полезен.
- Пусть $List(A)$ – тип список с элементами типа A.
- Пусть $Vec(A, n) = \{xs \in List(A) \mid length(xs) = n\}$ – тип списков длинны n.
- Тогда у нас есть биекция между множествами $List(A)$ и $\Sigma(n \in \mathbb{N})Vec(A, n)$.

Зависимые произведения

- Мы видели, что зависимые суммы обобщают произведение множеств $A \times B$.

Зависимые произведения

- Мы видели, что зависимые суммы обобщают произведение множеств $A \times B$.
- Зависимые произведения обобщают понятие функций $A \to B$.

Зависимые произведения

- Мы видели, что зависимые суммы обобщают произведение множеств $A \times B$.
- Зависимые произведения обобщают понятие функций $A \to B$.
- Пусть B_a – колекция множеств, зависящая от $a \in A$.

Зависимые произведения

- Мы видели, что зависимые суммы обобщают произведение множеств $A \times B$.
- Зависимые произведения обобщают понятие функций $A \to B$.
- Пусть B_a – колекция множеств, зависящая от $a \in A$.
- Тогда *зависимая функция* из A в B – это правило, которое каждому $a \in A$ сопоставляет элемент в B_a.

Определение

- Формально множество зависимых функций можно описать как подмножество множества функций $A \to \bigcup_{a \in A} B_a$, состоящее из функций f, таких что $f(a) \in B_a$ для любого $a \in A$.

Определение

- Формально множество зависимых функций можно описать как подмножество множества функций $A \to \bigcup_{a \in A} B_a$, состоящее из функций f, таких что $f(a) \in B_a$ для любого $a \in A$.
- Другой способ определить это множество – это через зависимые суммы:

$$\Pi(a \in A)B_a = \{f : A \to \Sigma(a \in A)B_a \mid \pi_1 \circ f = id_A\},$$

где $\pi_1 : \Sigma(a \in A)B_a \to A$ – это функция $\pi_1(a, b) = a$.

Определение

- Формально множество зависимых функций можно описать как подмножество множества функций $A \to \bigcup_{a \in A} B_a$, состоящее из функций f, таких что $f(a) \in B_a$ для любого $a \in A$.
- Другой способ определить это множество – это через зависимые суммы:

$$\Pi(a \in A)B_a = \{f : A \to \Sigma(a \in A)B_a \mid \pi_1 \circ f = id_A\},$$

где $\pi_1 : \Sigma(a \in A)B_a \to A$ – это функция $\pi_1(a, b) = a$.
- Пример: функцию *index* можно определить для любого n.

$$index : \Pi(n \in \mathbb{N})(\{i \in \mathbb{N} \mid i \leq n\} \to Vec(A, n) \to A)$$

Теорема о фактор-множестве и образе

- Пусть $f : A \to B$ – функция.

Теорема о фактор-множестве и образе

- Пусть $f : A \to B$ – функция.
- Тогда мы можем определить отношение эквивалентности на A как
$$a \sim a' \Leftrightarrow f(a) = f(a')$$

Теорема о фактор-множестве и образе

- Пусть $f : A \to B$ – функция.
- Тогда мы можем определить отношение эквивалентности на A как
$$a \sim a' \Leftrightarrow f(a) = f(a')$$
- Тогда существует биекция $h : A/{\sim} \to im(f)$, $h([a]_{\sim}) = f(a)$.

Теорема о фактор-множестве и образе

- Пусть $f : A \to B$ – функция.
- Тогда мы можем определить отношение эквивалентности на A как
$$a \sim a' \Leftrightarrow f(a) = f(a')$$
- Тогда существует биекция $h : A/{\sim} \to im(f)$, $h([a]_\sim) = f(a)$.
- В теории групп аналогом отношений эквивалентности являются нормальные подгруппы.

Теорема о фактор-множестве и образе

- Пусть $f : A \to B$ – функция.
- Тогда мы можем определить отношение эквивалентности на A как
$$a \sim a' \Leftrightarrow f(a) = f(a')$$
- Тогда существует биекция $h : A/{\sim} \to im(f)$, $h([a]_\sim) = f(a)$.
- В теории групп аналогом отношений эквивалентности являются нормальные подгруппы.
- Аналогом отношения $\sim$, определенного выше, является ядро гомоморфизма f.

Теорема о фактор-множестве и образе

- Пусть $f : A \to B$ – функция.
- Тогда мы можем определить отношение эквивалентности на A как
$$a \sim a' \Leftrightarrow f(a) = f(a')$$
- Тогда существует биекция $h : A/\!\sim \to im(f)$, $h([a]_\sim) = f(a)$.
- В теории групп аналогом отношений эквивалентности являются нормальные подгруппы.
- Аналогом отношения $\sim$, определенного выше, является ядро гомоморфизма f.
- Аналогом этой теоремы является теорма о существовании изоморфизма $A/Ker(f) \to im(f)$.

Следствия теоремы

- В случае групп у нас есть интересное следствие, что $|im(f)| = |A|/|Ker(f)|$. Есть ли аналогичное утверждение в случае множеств?

Следствия теоремы

- В случае групп у нас есть интересное следствие, что $|im(f)| = |A|/|Ker(f)|$. Есть ли аналогичное утверждение в случае множеств?
- Есть! Во-первых, заметим, что $|A| = \sum_{a \in A/\sim} |a|$.

Следствия теоремы

- В случае групп у нас есть интересное следствие, что $|im(f)| = |A|/|Ker(f)|$. Есть ли аналогичное утверждение в случае множеств?
- Есть! Во-первых, заметим, что $|A| = \sum_{a \in A/\sim} |a|$.
- Теперь из теоремы следует, что $|A| = \sum_{b \in im(f)} |h^{-1}(b)|$.

Следствия теоремы

- В случае групп у нас есть интересное следствие, что $|im(f)| = |A|/|Ker(f)|$. Есть ли аналогичное утверждение в случае множеств?
- Есть! Во-первых, заметим, что $|A| = \sum\limits_{a \in A/\sim} |a|$.
- Теперь из теоремы следует, что $|A| = \sum\limits_{b \in im(f)} |h^{-1}(b)|$.
- Если для любого $b \in im(f)$ множества $h^{-1}(b)$ равномощны, и их мощность равна k, то

$$|im(f)| = |A|/k$$

Следствия теоремы

- В случае групп у нас есть интересное следствие, что $|im(f)| = |A|/|Ker(f)|$. Есть ли аналогичное утверждение в случае множеств?
- Есть! Во-первых, заметим, что $|A| = \sum\limits_{a \in A/\sim} |a|$.
- Теперь из теоремы следует, что $|A| = \sum\limits_{b \in im(f)} |h^{-1}(b)|$.
- Если для любого $b \in im(f)$ множества $h^{-1}(b)$ равномощны, и их мощность равна k, то

$$|im(f)| = |A|/k$$

- Теперь утверждение для групп следует из этого замечания, так как в случае групп это верно.