Learn more

 $\{n\in\mathbb{Z}\mid (\exists k\in\mathbb{Z})[n=2k]\}$

expressed in set-builder notation.

The set of all even integers,

 (\mathbf{X})

The ...

Sound

Set-builder notation

Contents [hide]

3 More complex expressions on the left side of the notation

From Wikipedia, the free encyclopedia

Sets defined by enumeration

2.1 Specifying the domain

4 Equivalent predicates yield equal sets

2 Sets defined by a predicate

2.2 Examples

5 Set existence axiom

6 In programming languages

Talk

Article

We usually invite the world to create the sum of all human knowledge. Now, we are inviting the world to create the sound of all human knowledge.

In set theory and its applications to logic, mathematics, and computer science, set-builder notation is a mathematical notation for describing a set

Search Wikipedia Edit | View history [Help with translations!]

Read

Main page Contents Current events Random article **About Wikipedia** Contact us

Donate Contribute Help Learn to edit Community portal

Recent changes

Upload file

Tools What links here Related changes Special pages Permanent link Page information

Cite this page Wikidata item Print/export

Download as PDF

Printable version Languages

বাংলা 한국어 Bahasa Indonesia Íslenska Русский Українська 中文

Edit links

Sets defined by enumeration [edit]

7 See also

8 Notes

A set can be described directly by enumerating all of its elements between curly brackets, as in the following two examples:

 $\{7, 3, 15, 31\}$ is the set containing the four numbers 3, 7, 15, and 31, and nothing else.

Defining sets by properties is also known as **set comprehension**, **set abstraction** or as defining a set's **intension**.

 $\{a,c,b\}=\{a,b,c\}$ is the set containing a,b, and c, and nothing else (there is no order among the elements of a set).

by enumerating its elements, or stating the properties that its members must satisfy. [1]

This is sometimes called the "roster method" for specifying a set. [2]

When it is desired to denote a set that contains elements from a regular sequence, an ellipses notation may be employed, as shown in the next examples:

 $\{1, 2, 3, \ldots, 100\}$ is the set of integers between 1 and 100 inclusive.

 $\{1, 2, 3, \ldots\}$ is the set of natural numbers.

 $\{\ldots,-2,-1,0,1,2,\ldots\}=\{0,1,-1,2,-2,\ldots\}$ is the set of all integers.

There is no order among the elements of a set (this explains and validates the equality of the last example), but with the ellipses notation, we use an ordered sequence before (or

after) the ellipsis as a convenient notational vehicle for explaining which elements are in a set. The first few elements of the sequence are shown, then the ellipses indicate that the simplest interpretation should be applied for continuing the sequence. Should no terminating value appear to the right of the ellipses, then the sequence is considered to be unbounded.

In general, $\{1,\ldots,n\}$ denotes the set of all natural numbers i such that $1 \leq i \leq n$. Another notation for $\{1,\ldots,n\}$ is the bracket notation [n]. A subtle special case is n=0, in which $[0]=\{1,\ldots,0\}$ is equal to the empty set \emptyset . Similarly, $\{a_1,\ldots,a_n\}$ denotes the set of all a_i for $1\leq i\leq n$.

In each preceding example, each set is described by enumerating its elements. Not all sets can be described in this way, or if they can, their enumeration may be too long or too complicated to be useful. Therefore, many sets are defined by a property that characterizes their elements. This characterization may be done informally using general prose, as in the

However, the prose approach may lack accuracy or be ambiguous. Thus, set-builder notation is often used with a predicate characterizing the elements of the set being defined, as

{ addresses on Pine Street } is the set of all addresses on Pine Street.

values of x that satisfy the formula Φ .^[4] It may be the empty set, if no value of x satisfies the formula.

described in the following section. Sets defined by a predicate [edit]

Set-builder notation can be used to describe a set that is defined by a predicate, that is, a logical formula that evaluates to true for an element of the set, and false otherwise. [3] In this

form, set-builder notation has three parts: a variable, a colon or vertical bar separator, and a predicate. Thus there is a variable on the left of the separator, and a rule on the right of it. These three parts are contained in curly brackets:

or $\{x:\Phi(x)\}.$

for which the predicate holds (is true) belong to the set being defined. All values of x for which the predicate does not hold do not belong to the set. Thus $\{x \mid \Phi(x)\}$ is the set of all

The vertical bar (or colon) is a separator that can be read as "**such that**", "for which", or "with the property that". The formula $\Phi(x)$ is said to be the *rule* or the *predicate*. All values of x

 $\{x\mid \Phi(x)\}$

following example.

A domain E can appear on the left of the vertical bar:^[5]

Specifying the domain [edit]

 $\{x\in E\mid \Phi(x)\},\$

or by adjoining it to the predicate: $\{x\mid x\in E ext{ and } \Phi(x)\} \quad ext{or} \quad \{x\mid x\in E \wedge \Phi(x)\}.$

The \in symbol here denotes set membership, while the \land symbol denotes the logical "and" operator, known as logical conjunction. This notation represents the set of all values of xthat belong to some given set E for which the predicate is true (see "Set existence axiom" below). If $\Phi(x)$ is a conjunction $\Phi_1(x) \wedge \Phi_2(x)$, then $\{x \in E \mid \Phi(x)\}$ is sometimes

written $\{x\in E\mid \Phi_1(x),\Phi_2(x)\}$, using a comma instead of the symbol \wedge . In general, it is not a good idea to consider sets without defining a domain of discourse, as this would represent the subset of all possible things that may exist for which the predicate is true. This can easily lead to contradictions and paradoxes. For example, Russell's paradox shows that the expression $\{x \mid x \notin x\}$, although seemingly well formed as a set builder

expression, cannot define a set without producing a contradiction.^[6] In cases where the set E is clear from context, it may be not explicitly specified. It is common in the literature for an author to state the domain ahead of time, and then not specify it in the set-builder notation. For example, an author may say something such as, "Unless otherwise stated, variables are to be taken to be natural numbers," though in less formal

contexts where the domain can be assumed, a written mention is often unnecessary. **Examples** [edit]

The following examples illustrate particular sets defined by set-builder notation via predicates. In each case, the domain is specified on the left side of the vertical bar, while the rule is specified on the right side.

formula on the right.

 $\{x \in \mathbb{R} \mid x > 0\}$ is the set of all strictly positive real numbers, which can be written in interval notation as $(0, \infty)$. $\{x \in \mathbb{R} \mid |x|=1\}$ is the set $\{-1,1\}$. This set can also be defined as $\{x \in \mathbb{R} \mid x^2=1\}$; see equivalent predicates yield equal sets below.

For each integer m, we can define $|G_m|=\{x\in\mathbb{Z}\mid x\geq m\}=\{m,m+1,m+2,\ldots\}$. As an example, $|G_3|=\{x\in\mathbb{Z}\mid x\geq 3\}=\{3,4,5,\ldots\}$ and

 $G_{-2} = \{-2, -1, 0, \ldots\}$. $\{(x,y) \in \mathbb{R} \times \mathbb{R} \mid 0 < y < f(x)\}$ is the set of pairs of real numbers such that y is greater than 0 and less than f(x), for a given function f. Here the cartesian product $\mathbb{R} \times \mathbb{R}$

denotes the set of ordered pairs of real numbers.

 $\{n \in \mathbb{N} \mid (\exists k)[k \in \mathbb{N} \land n = 2k]\}$ is the set of all even natural numbers. The \land sign stands for "and", which is known as logical conjunction. The \exists sign stands for "there exists", which is known as existential quantification. So for example, $(\exists x)P(x)$ is read as "there exists an x such that P(x)". $\{n \mid (\exists k \in \mathbb{N}) [n=2k]\}$ is a notational variant for the same set of even natural numbers. It is not necessary to specify that n is a natural number, as this is implied by the

 $\{a \in \mathbb{R} \mid (\exists p \in \mathbb{Z})(\exists q \in \mathbb{Z})[q \neq 0 \land aq = p]\}$ is the set of rational numbers; that is, real numbers that can be written as the ratio of two integers.

More complex expressions on the left side of the notation [edit] An extension of set-builder notation replaces the single variable x with an expression. So instead of $\{x \mid \Phi(x)\}$, we may have $\{f(x) \mid \Phi(x)\}$, which should be read

 $\{f(x)\mid \Phi(x)\}=\{y\mid \exists x(y=f(x)\wedge \Phi(x))\}$. For example: $\{2n \mid n \in \mathbb{N}\}$, where \mathbb{N} is the set of all natural numbers, is the set of all even natural numbers.

 $\{p/q\mid p,q\in\mathbb{Z},q
eq 0\}$, where \mathbb{Z} is the set of all integers, is \mathbb{Q} , the set of all rational numbers. $\{2t+1 \mid t \in \mathbb{Z}\}$ is the set of odd integers. $\{(t,2t+1)\mid t\in\mathbb{Z}\}$ creates a set of pairs, where each pair puts an integer into correspondence with an odd integer.

When inverse functions can be explicitly stated, the expression on the left can be eliminated through simple substitution. Consider the example set $\{2t+1\mid t\in\mathbb{Z}\}$. Make the substitution u=2t+1, which is to say t=(u-1)/2, then replace t in the set builder notation to find

Equivalent predicates yield equal sets [edit]

Two sets are equal if and only if they have the same elements. Sets defined by set builder notation are equal if and only if their set builder rules, including the domain specifiers, are equivalent. That is

 $\{2t+1 \mid t \in \mathbb{Z}\} = \{u \mid (u-1)/2 \in \mathbb{Z}\}.$

 $(orall t)[(t\in A\wedge P(t))\Leftrightarrow (t\in B\wedge Q(t))]$. Therefore, in order to prove the equality of two sets defined by set builder notation, it suffices to prove the equivalence of their predicates, including the domain qualifiers. For example,

 $\{x\in A\mid P(x)\}=\{x\in B\mid Q(x)\}$

if and only if

lists.

Python

 $\{x\in\mathbb{R}\mid x^2=1\}=\{x\in\mathbb{Q}\mid |x|=1\}$ because the two rule predicates are logically equivalent:

Set existence axiom [edit]

 $(x\in \mathbb{R}\wedge x^2=1)\Leftrightarrow (x\in \mathbb{Q}\wedge |x|=1).$

In many formal set theories, such as Zermelo-Fraenkel set theory, set builder notation is not part of the formal syntax of the theory. Instead, there is a set existence axiom scheme, which states that if E is a set and $\Phi(x)$ is a formula in the language of set theory, then there is a set Y whose members are exactly the elements of E that satisfy Φ : $(orall E)(\exists Y)(orall x)[x\in Y\Leftrightarrow x\in E\wedge \Phi(x)].$

This equivalence holds because, for any real number x, we have $x^2 = 1$ if and only if x is a rational number with |x| = 1. In particular, both sets are equal to the set $\{-1, 1\}$.

In programming languages [edit] Main article: List comprehension A similar notation available in a number of programming languages (notably Python and Haskell) is the list comprehension, which combines map and filter operations over one or more

Example 1

The set Y obtained from this axiom is exactly the set described in set builder notation as $\{x \in E \mid \Phi(x)\}$.

{l for l in L}

syntax. Haskell replaces the set-builder's braces with square brackets and uses symbols, including the standard set-builder vertical bar. The same can be achieved in Scala using Sequence Comprehensions, where the "for" keyword returns a list of the yielded variables using the "yield" keyword. [7] Consider these set-builder notation examples in some programming languages:

Example 2

In Python, the set-builder's braces are replaced with square brackets, parentheses, or curly braces, giving list, generator, and set objects, respectively. Python uses an English-based

 $\{(k,x)\mid k\in K\wedge x\in X\wedge P(x)\}$ $\{l \mid l \in L\}$ Set-builder

 $\{(k, x) \text{ for } k \text{ in } K \text{ for } x \text{ in } X \text{ if } P(x)\}$

Haskell	[l l <- ls]	[(k, x) k <- ks, x <- xs, p x]
Scala	for (l <- L) yield l	for (k <- K; x <- X if P(x)) yield (k,x)
C#	from l in L select l	<pre>from k in K from x in X where P(x) select (k,x)</pre>
SQL	SELECT l FROM L_set	SELECT k, x FROM K_set, X_set WHERE P(x)
Prolog	<pre>setof(L,member(L,Ls),Result)</pre>	<pre>setof((K,X),(member(K,Ks),member(X,Xs),call(P,X)),Result)</pre>
Ruby	L.map{ l l}	<pre>K_*product(X)_*select{ k,x P(x) }</pre>
Erlang	[l l <- ls]	
Julia	[l for l ∈ L]	[(k, x) for $k \in K$ for $x \in X$ if $P(x)$]

Glossary of set theory

Set theory

[show]

Notes [edit] 1. A Rosen, Kenneth (2007). Discrete Mathematics and its Applications (6th ed.). New York, NY: McGraw-Hill. pp. 111–112. ISBN 978-0-07-288008-3.

3. ^ Michael J Cullinan, 2012, A Transition to Mathematics with Proofs, Jones & Bartlett, pp. 44ff.

4. * Weisstein, Eric W. "Set" . mathworld.wolfram.com. Retrieved 20 August 2020. 5. * "Set-Builder Notation" . mathsisfun.com. Retrieved 20 August 2020. 6. Irvine, Andrew David; Deutsch, Harry (9 October 2016) [1995]. "Russell's Paradox". Stanford Encyclopedia of Philosophy. Retrieved 6 August 2017.

2. A Richard Aufmann, Vernon C. Barker, and Joanne Lockwood, 2007, *Intermediate Algebra with Applications*, Brooks Cole, p. 6.

V •T •E

Categories: Set theory | Mathematical notation

7. ^ "Sequence Comprehensions" . Scala. Retrieved 6 August 2017.

This page was last edited on 24 September 2022, at 23:10 (UTC).

Text is available under the Creative Commons Attribution-ShareAlike License 3.0; additional terms may apply. By using this site, you agree to the Terms of Use and Privacy Policy. Wikipedia® is a registered trademark of the Wikimedia Foundation, Inc., a non-profit organization.

Privacy policy About Wikipedia Disclaimers Contact Wikipedia Mobile view Developers Statistics Cookie statement