Set-builder notation

From Wikipedia, the free encyclopedia

$$\{a \mid \exists p, q \in \mathbf{Z}\ (q \neq 0 \land aq = p)\}$$

The set of all rational numbers, expressed in set-builder notation.

In set theory and its applications to logic, mathematics, and computer science, **set-builder notation** is a mathematical notation for describing a set by stating the properties that its members must satisfy.[1] Forming sets in this manner is also known as **set comprehension**, **set abstraction** or as defining a set's **intension**. Although some simply refer to it as *set notation,* that label may be better reserved for the broader class of means of denoting sets.

Contents

Direct, ellipses, and informally specified sets

A set is an unordered list of *elements*. (An *element* may also be referred to as a *member*). An element may be any mathematical entity.

We can denote a set directly by listing all of its elements between curly brackets, as in the following two examples:

- $\{7, 3, 15, 31\}$ is a set holding the four numbers 3, 7, 15, and 31.
- $\{a, c, b\}$ is the set containing 'a','b', and 'c'.

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 two examples.

- $\{1, 2, 3, \ldots, 100\}$ is the set of integers between 1 and 100 inclusive.
- $\{0, 1, 2, \ldots\}$ is the set of natural numbers.

There is no order among the elements of a set, but with the ellipses notation we show an ordered sequence before the ellipsis as a convenient notational vehicle for explaining to a reader which members 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.

Mathematicians sometimes denote a set using general prose, as shown in the following example.

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

The meaning of this prose must be clear to the reader or the mathematician who wrote it has failed to sufficiently define the set for the reader.

The ellipses and simple prose approaches give the reader rules for building the set rather than directly presenting the elements. Mathematicians find this approach of providing building rules to be convenient and important so they have extended and formalized the set builder notation as further described in this article.

Formal set builder notation sets

A set denoted in formal set builder notation has three parts, a variable, a colon or vertical bar separator, and a logical 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.

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

or

$$\{x \,:\, \Phi(x)\}$$

The *x* is taken to be a variable. The vertical bar, or colon, separator is read as 'such that'. Φ(*x*) is said to be the 'rule' or the 'set builder rule'. It is a logical predicate that evaluates to 'true' or 'false'. All values of *x* where the rule predicate is true belong in the set. All values of *x* where the rule is false are not in the set.

More formally, we say that Φ(*x*) is a formula in predicate logic where the variable *x* is universally quantified (see quantifiers) over the domain for the rule. Then in the universe of discourse values of x where Φ(*x*) is true are in the set, and those where Φ(*x*) are false are not in the set. We may also say that the set members are the extension of Φ(*x*), or that set builder notation binds the variable *x* to the rule.

Here are some examples of set builder notation in action:

- $\{x \mid x \in \mathbf{R} \wedge x = x^2\}$ is the set $\{0, 1\}$,
- $\{x \mid x \in \mathbf{R} \wedge x > 0\}$ is the set of all positive real numbers.
- $\{(x, y) \mid 0 < y < f(x)\}$ is the set of (x,y) such that y is greater than 0 and less than f(x).
- $\{k \mid \exists n \in \mathbf{N}\ k = 2n\}$ is the set of all even natural numbers,

- $\{a \mid \exists p, q \in \mathbf{Z}\ (q \neq 0 \land aq = p)\}$ is the set of rational numbers; that is, numbers that can be written as the ratio of two integers.
- $\mathbf{N}_m = \{x \mid x \in \mathbf{Z} \land x \geq m\} = \{m, m+1, m+2, \ldots\}$, where *m* is some integer. Thus, e.g., $x_1 = m,\ x_2 = m+1$, etc. (n.b.: in the case of sets, the order is not important; $x_1 = m+2$ could be used). As an example, $\mathbf{N}_3 = \{x \mid x \in \mathbf{Z} \land x \geq 3\} = \{3, 4, 5, \ldots\}$.

The '∈' symbol denotes set membership, and can be read as 'element of', 'member of', 'is in', 'belongs to', or 'lies in'. When used in logic a clause of the form *x* ∈ {1, 2, 3} is either true or false depending if *x* is one of the values in the set. When used for quantification such a clause means that *x* ranges over the values 1, 2, or 3.

The $\land$ sign stands for *"and"* or *"conjunction"*. This binary operator requires that both clauses to the left of it and to the right of it to be 'true' for the entire clause to be true. A related connector is $\lor$ which stands for *or* or *disjunction*.

Sometimes multiple rules are given separated by a comma or semicolon, in such case we take the rules in conjunction, i.e. we interpret the comma or semicolon to mean the same as $\land$.

The ∃ sign stands for "there exists" and is formally known as existential quantification. Quantification takes a variable and a predicate, and evaluates to true or false. So for example, ∃x:P(x) reads 'there exists an x for which P(x) is true'. If such an x does exist, then ∃x:P(x) is true, otherwise it is false. Another common quantifier is ∀, universal quantification. ∀x:P(x) will be true if for all values of x P(x) is true, which is to say there does not exist an x where P(x) is false, ¬∃x:¬P(x).

Expressions to the left of 'such that' rather than a variable

An extension of set-builder notation replaces the single variable *x* with a term *T* that may include one or more variables, combined with functions acting on them. So instead of {*x* : Φ(*x*)}, we have {*T* : Φ(x_1 ... x_n)}, where *T* is a term involving variables x_1 through x_n. For example:

- $\{2n : n \in N\}$, where **N** is the set of all natural numbers, is the set of all even natural numbers.
- $\{p/q : p, q \in Z, q \neq 0\}$, where **Z** is the set of all integers, is the set of all rational numbers (**Q**).
- $\{2t + 1 \mid t \in \mathbf{Z}\}$ is the set of odd integers.
- $\{(t, 2t+1) \mid t \in \mathbf{Z}\}$ creates a set of pairs, where each pair puts an integer into correspondence with an odd integer.
- $\{2x + 1 = 5 \mid x \in \mathbf{N}\}$ is the set $\{true, false\}$ because the expression $2x + 1 = 5$ evaluates to either true or false given the various natural numbers.
- $\{x \in R \mid x \in \mathbf{C} \land n \in \mathbf{N} \land x = \pi n\}$ where *C* is the set of complex numbers, is the set $\{true\}$.

Note in this last example the $x \in \mathbf{R}$ appears to the left of the 'such that' so it is evaluated as an expression. Thus it is true when *x* is in *R*, and it is false when *x* is not in *R*. This follows from our definition of set builder notation and the extension here to allow expressions. Be careful when reading such expressions as there is a common notational convention where set inclusion found on the left should instead be interpreted as a domain quantifier for the variable. See the description of that convention in the section below.

When inverse functions are available the expression on the left can be eliminated through simple substitution. Consider the example set above {2 *t*+ 1 | t ∈ **Z**}. Make the substitution, *u* = 2 *t*+ 1, resulting in *t* = (*u*- 1)/2, then replace *t* and the expression to find:

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

Convention of annotating the variable domain on the left of the 'such that'

There is a convention found in the literature where set membership clauses are found to the left of the 'such that', but they are intended to annotate the domain the affected variable belongs to rather than to be understood as part of the expression. Here is an example:

- $\{x \in \mathbf{R} \mid x = x^2\}$ is the set $\{0, 1\}$,

In this case the left side expression is just *x*, and the set membership statement is to be taken as part of the rule. In our formal set builder notation this would be written as:

- $\{x \mid x \in \mathbf{R} \land x = x^2\}$ is the set $\{0, 1\}$

It is typically clear from context when this shorthand convention is being used. However an author must clarify in any situation where there might be ambiguity between domain quantification or membership operation . When doing proofs on sets that follow this convention it is important to first move the domain qualifiers back to the rule predicates, or if they are common to all rules, they may be moved into universal quantifiers.

Leaving the variable domain understood by context

It is common in the literature for an author to up front universally quantify variable domains and then not state them in the rule predicates. An author may say something such as, "Unless otherwise stated variables are to be taken as Natural Numbers."

Taking one of our sets from the first section as an example, we can say, "The universe of discourse can be taken to be the set of real numbers, where not specified inside the notation," and then write:

- $\{x \mid x = x^2\}$ is the set $\{0, 1\}$

In such situations when doing proofs, the understood domain is included in a universal quantifier.

Equivalent builder predicates means equivalent sets

Two sets are equal if and only if their set builder rules, including the domain specifiers, are logically equivalent. For example, $\{x|x \in \mathbf{R} \wedge x^2 = 1\} = \{x|x \in \mathbf{R} \wedge \ |x| = 1\}$ because, the two rule predicates are logically equivalent:
$(x \in \mathbf{R} \wedge x^2 = 1) \Leftrightarrow (x \in \mathbf{R} \wedge \ |x| = 1)$

That is to say, that for any real number *x*, x^2 = 1 will be a true statement if and only if for real number x, |*x*| = 1 is a true statement.

We can state this result more formally by considering two generic sets, namely, the set of elements created from set builder predicate *P*,

- $A = \{x|P(x)\}$,

and the set of elements created by set builder predicate *Q*,

- $B = \{x|Q(x)\}$.

Then sets A and B will be equal if

- $\forall t : P(t) \Leftrightarrow Q(t)$.

The inverse situation is also true, i.e. if the two sets have the same members, then their set builder rule predicates are logically equivalent. Hence we can say in general:

$$(\forall_x : P(x) \Leftrightarrow Q(x)) \Leftrightarrow (\{x|P(x)\} = \{x|Q(x)\})$$

Russell's Paradox

Consider set $R = \{S \mid S \notin S\}$ defined to be the set of all sets *S* that do not contain themselves.

Let's ask a question about *R*. Does this set contain itself? I.e. can it be one of the elements *S*?

If *R* does not contain itself, then according to the set builder rule it fits the criteria for being an *S* element, so it should be in *R*; however, if it is in *R* then it contains itself! We arrive at a contradiction.

Now consider the case that *R* contains itself, then by definition it should not be in the set *R*. Another contradiction!

According to the constructs of Whitehead's set theory, all elements are either in a set, or not in a set, but here using the same theory, Bertrand Russell shows an example of element, *R* which can not be either. This inconsistency is known as Russell's Paradox.

It is possible to avoid this paradox by restricting the richness in expressive power of the original set theory. To illustrate this in terms of our notation, let $X = \{x \mid x \in A \wedge P(x)\}$ denotes the set of every element of *A* satisfying the predicate *P*(*x*). The canonical restriction on set builder notation asserts that *X* is a set only if *A* is already known to be a set. This restriction is codified in the axiom schema of separation present in standard axiomatic set theory. Note that this axiom schema excludes *R* from sethood.

Z notation

In Z notation, the set of all *x* (in a universe of discourse *A*) satisfying the condition *P*(*x*) would be written $\{x : A \mid P(x)\}$. In Z, an element x's set membership is written as $(x : A)$ instead of $(x \in A)$; the vertical bar is used to indicate a predicate. Versions of set builder notation are also available in Z which allow for terms more complicated than a single variable, using a bullet to indicate the form of members of the set. So $\{x : A \mid P(x) \bullet F(x)\}$ denotes the set of all values *F*(*x*), where *x* is in *A* and *P*(*x*) holds.

Parallels in programming languages

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 lists.

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 syntax. Haskell replaces the set-builder's braces with square brackets and uses symbols, including the standard set-builder vertical bar. Consider these examples given in set-builder notation, Python, and Haskell:

	Example 1	**Example 2**
Set-builder	$\{l \mid l \in L\}$	$\{(k, x) \mid k \in K \wedge x \in X \wedge P(x)\}$
Python	`{l for l in L}`	`{(k, x) for k in K for x in X if P(x)}`
Haskell	`[l | l <- ls]`	`[(k, x) | k <- ks, x <- xs, p x]`

The set builder notation and list comprehension notation are both instances of a more general notation known as *monad comprehensions*, which permits map/filter-like operations over any monad with a zero element.[2]

References

1. ^ Rosen, Kenneth (2007). *Discrete Mathematics and its Applications* (6th ed.). New York, NY: McGraw-Hill. pp. 111–112. ISBN 978-0-07-288008-3.
2. ^ Nils Schweinsberg (27 Nov 2010). "Fun with monad comprehensions" (http://blog.n-sch.de/2010/11/27/fun-with-monad-comprehensions/). Retrieved 4 July 2011.

Categories: Set theory | Mathematical notation