Re: Sets question (maybe)

Source: http://sci.tech−archive.net/Archive/sci.math/2007−10/msg04211.html

---

- *From*: "cbrown@xxxxxxxxxxxxxxxxx" <cbrown@xxxxxxxxxxxxxxxxx>
- *Date*: Tue, 23 Oct 2007 10:44:55 −0700

On Oct 23, 7:22 am, Name And Address Supplied
<name_and_address_suppl...@xxxxxxxxxxx> wrote:

> Dear all,
>
> I wonder if you can help with the following problem. I'm not a mathematician, so please bear with me.
>
> Informally speaking: I have a set of, say, 5 entities, and I wish to give each a unique index j. In particular, I want to pick one entity −− it doesn't matter which −− and I want to designate that j = 1. Then I want to pick another, and index that one j = 2. And so on until I have five individuals indexed 1 to 5.

Note that this is the same thing as assigning some element of the 5 entities to each of 1,2,3,4,5.

And that's the same thing as defining a function S with S(1) = (the element of from the 5 entities you picked first), S(2) = (the element of J you picked second), etc.

So in a way, you are talking about a /function/: a special sort of function with domain {1,2,3,4,5} and range the 5 entities.

> Now I want to order this list,
> so that it runs from 1 through to 5.

Isn't it /already/ ordered by observing that, for example, 1 < 2, 2 < 3, etc.?

> Q1: What do I call this list?

In mathematics, it's usually referred to as a finite sequence (or just a list).

If J is a sequence, whose elements are taken from the set S, we will usually write J = (j_1, j_2, j_3, ..., j_n), where n is the "number of entities"; and where "j_1" is how we write "j with subscript 1" in sci.math. Each j_i is some member of S.

Thinking of J as a function, J(1) = j_1, J(2) = j_2, etc.

Note that in mathematics, a list can also refer to an /infinite/ sequence (j_1, j_2, ..., j_n, ...).

> I think it is not a set in the strict
> sense, as a set is not ordered, and also a set with repeated elements
> is equal to the same set with the repeated elements removed.

That is correct. The standard way to approach this issues is as follows:

{x, y} = {y, x} as sets: two sets are th same if, and only if, they have the same members.

So we instead /define/ (x, y) to be the /set/ {x, {x, y}}. This is "the ordered pair (x,y)". A little thought (well, plus the fact that it is forbidden for a set to have itself as a member) should convince you that (x, y) = (y, x) if, and only if, x = y.

So we can then think of the finite /sequence/ J = (j_1, j_2, ..., j_5) as the /set/ of ordered pairs S = {(1, j_1), (2, j_2), ..., (5, j_5)}.

This is exactly the same (in set theory) as defining J to be a / function/ which takes the naturals {1,2,3,4,5} (often just written as "the set 5") to the set J. So we can also write "J : 5 −> S", and talk about J(3) being the third element of the sequence J, etc.

> But I
> want to describe the object J = {1, 2, 3, 4, 5}...

Careful! Is J the set of /entities/ (whatever they might be)? The above sort of confuses J with the set of /indexes/...

> , and say things like
> "j E J" (j is a member of J).

We might loosely say "j is a member of the sequence J = (j_1, j_2, j_3, ..., j_n)" to mean the more rigorously correct "there exists i in (1, 2, .., n} such that j = j_i", or equivalently (given the above definitions) "... such that (i, j) in J" or "... such that j = J(i)".

> I also want to say that the jth element of J is j...

Careful − you're again confusing your /indexes/ with your /entities/!
It is also possible that j is /not/ the jth element of J.

> −− this of course would not be true if J wasn't ordered / had repeated elements. So, what is J? A set, a list, a group, what?

It's a sequence or list, which is a special sort of function (it has a range which is from 1 to some natural), which is a special sort of set.

> Q2: I don't know that there are actually 5 entities, but I do know there is a finite number of them. I want to index them as above, and I want to say in a mathematically compact way that the indices are the natural numbers running from 1 to the maximum index value (which is equal to the number of entities I want to index). In particular, I don't want to have to assign a symbol to denote the total number of entities.

If S is the set of entities, one often writes |S| (which is spoken as "the cardinality of J" or "card of J") to refer to the number of (distinct!) elements in S. Then you can talk about the finite sequence
J = {j_1, j_2, ..., j_|S|}; or equivalently consider J as a function
J : |S| −> S, with j_i = J(i).

> Q3: If J were a set, and K = {1,2} were some other set, I could define a Cartesian product JxK = {{{1,1},{1,2}},{{2,1},{2,2}},{{3,1},{3,2}}, {{4,1},{4,2}},{{5,1},{5,2}}}.

That's /not/ the Cartesian product J x K. The Cartesian product is the /set/ of all /ordered pairs/ of the form (j,k), j in J, k in K. It has |J|*|K| = 10 elements. What you have written above has elements of the form {{j, 1}, {j, 2}} for j in J. It has only |J| = 5 elements;
each such element having |K| = 2 elements.

> But if J and K are not sets, but are the
> type of objects I've described above, can I still make this operation?

Without a bit more context, it's not clear exactly /what/ it is you're trying to invoke with this operation. My guess is that, given natural numbers a, b, you want to be able to talk about the ordered pair (j_a, k_b), where j_a is the "a'th" element in the sequence J, and k_b is the "b'th" element of the sequence K.

You could express that as a function T which maps each possible pair (a,b) to the corresponding pair (j_a, k_b). In that case (recalling that J and K are functions in the set theory sense) you can just define the function T(a,b) = (J(a), K(b)).

As a /set/, T would then have elements of the form ((a,b), (J(a),K(b)); an ordered pair of ordered pairs.

Is that what you're trying to formalize?

> Or is there an analogous operation, and if so, what is it called?
>
> Q4: Is the "curly bracket" notation okay for what I am doing, i.e. "{"
> and "}"?

No. You want to use "(" and ")" to indicate that you are talking about a finite sequence. Sometimes people alternatively use "<" and ">" if there are lots of other parentheses kicking around already; for example J = <j_1, j_2, j_3, ..., j_n>.

Cheers − Chas

.