# Re: Constructibility of X −> X^2 bijection

***Source:*** http://sci.tech−archive.net/Archive/sci.math/2007−07/msg04700.html

- *From*: i.am.a.whiz@xxxxxxxxxxxxxxxx
- *Date*: Fri, 27 Jul 2007 21:08:23 −0000

Robert Israel brilliantly explained:

> Without any information about X except that it is not finite, how could you explicitly specify any non−constant function on X? You can't do it without some way of distinguishing members of X from each other.

I agree completely, except for the nitpick that "non−constant" isn't precisely correct here. Consider the mapping from the primitive element x to the ordered tuple <x,x>. That isn't a constant function, nor even an identity mapping, yet is definable universally, whereby you state that definition first, and then it works for any set X whatsoever (to the set XxX).

I suspect it would require a nontrivial proof to establish that the only kinds of mappings you can define universally (from arbitrary domain set) are constant functions (but then you need a specific target element, which can't be in the original set at all for obvious reasons) and pseudo−identity mappings (x −> <x,x> as above, and also x −> <const,x> and x −> <x,const> where const would be in some X's but not other X's), none of which are capable of covering the entire XxX set even when const is in X.

Now if you modify the original premise to assume that the set X has been explicitly constructed, i.e. there's some grammar that enumerates exactly the elements of X (or a grammar that enumerates a larger set combined with a computable predicate which restricts the generated set down to the desired set), with the further premises that the grammar uses only a finite alphabet, and the alphabet is totally ordered, and only finite strings from the grammar are allowed, then I think we can write a formal definition of the mapping before the set X is known. We simply use the total order on the alphabet together with considerations of length and lexicographic order to produce a total ordering of all the ways that the grammar can generate anything, and when two different grammatical constructions can produce the same element of X we choose the earliest of them. That produces a canonical ordering of X per that particular grammar. Then use Cantor's usual zigzag rule to map X −> XxX per that canonical ordering. But note, this definition is in terms of the particular ordered−grammar, not in terms of the set X per se. For exactly the same set X, two different grammars will yield the same set, which yields two different canonical orderings on X, hence two different Cantor−zigzag mappings. For example, simply swap the sequence of the first two characters in the alphabet. But even if we avoid that trivial counter−example by presuming a universal ordering for all possible characters in all possible grammars, still we might find two non−ismorphic grammars which happen to generate the same set X.

As to why I bother with grammars, consider the constructable subset of the real numbers. Consider a set−theoretic grammar whereby we can define specific Dedekind cuts (as sets of rational numbers) or Cauchy sequences (as mappings from positive integers to rational numbers). For example {x : x^3 < 7} defines the cube root of 7 as a Dedekind cut. Now that expression contains 13 characters, so in the length−lexicographic ordering it would come before {x : x*5 < 29} which contains 14 characters, but after the simple {x : x < 5} which contains only 11 characters. Now imagine a different grammar that didn't have the ^ operator, and instead required repeated multiplication for constant−exponent expressions, or some really complicated set−theoretic expression for variable exponents. That would force the cube root of 7 to be expressed in more than 14 characters, changing the ordering of the expressions, thus changing the Cantor−zigzag mapping. Note that the set X does not contain these character expressions, but instead contains the *numbers* defined by these expressions. So there's a counterexample where exactly the same set X of constructable real numbers is generated by two different grammars yielding two different orderings on X hence two different Cantor−zigzag mappings from X to XxX.

Now I have a question: Is it possible to assign a total ordering on all grammars, and thereby pick the canonical grammar for generating the constructable real numbers? I think this would require we know a canonical ordering for the union of all characters that can ever appear in any such grammar, but with that caveat taken for granted I think it could work. If so, then at least for constructable per finite grammar (hence countable) infinite sets we may have a universal way to define the desired mapping.

But the full question asked by the OP, no way, per your explanation.
.