

Re: A Definition of an Algorithm

Source: <http://sci.tech-archive.net/Archive/sci.math.research/2006-02/msg00150.html>

- *From:* noson@xxxxxxxxxxxxxxxxxxxxxxxx
 - *Date:* Sun, 26 Feb 2006 16:00:15 +0000 (UTC)
-

From: george

Nowadays, in the most usual set theories, a natural number is NOT some equivalence class with more members THAN THERE ARE natural numbers, or even than there are sets. Nowadays, small natural numbers are SMALL things represented by SMALL sets (the smaller the number, the smaller the set). That whole approach is just fundamentally backwards. It invites you to think that the class of sets-of-size-3 has to be understood BEFORE what "3" is understood. That cannot factually be the case. In real life, you have to understand what 3 is, FIRST, IN ORDER to tell what sets to put into the equivalence class and which to leave out. The same is obviously going to be true of "algorithms".

Frege did not define A natural number. He defined the SET of natural numbers.

So too, I am not defining An algorithm. I am defining the SET of algorithms.

The set of all programs is partitioned into equivalence classes. Two programs are equivalent if they are ``essentially" the same program.

I'm going to guess that means they "perform" the same mapping of input to output. The textbook did begin by stressing that that was necessary (even though it is obviously not sufficient).

You guessed wrong!

The set of all equivalence
classes is the category of all algorithms.

If you want a category–theoretic treatment (as opposed to a set–theoretic treatment) of algorithms, then surely the best first place to look for inspiration would be to a category–theoretic treatment of the natural numbers. Have you found one?

Yes it is a natural number object in a category. Nothing to do with this.

In order to explore these ideas, the set of primitive recursive functions is considered. Each primitive recursive function can be described by many labeled binary trees that show how the function is built up.

And in a great many other ways as well.
JEEzus! If THIS is all you are going to do, then can't you see that ALL you are doing is just INVENTING ANOTHER PROGRAMMING language?!? SURELY the WHOLE point to THIS enterprise, if it can have a point, is to achieve INDEPENDENCE from a particular programming language! Or are you saying that a language built with binary trees is fundamentally different from a linear/verbal/syntactic language? I'm sorry, you can't defend THAT distinction; the mere existence of the phrase "parse tree" (for compiling the verbal/syntactic languages) proves THAT.

On page 36 of the paper, I give a "language independant definition of the category of algorithms.

That is an irrelevant misuse of category theory.

Re: A Definition of an Algorithm

Why?

All the best,
Noson

.