ALGORITHM METHODS
Thomas C. Folsom

It is very easy to be blinded to the essential uselessness of [current patent law doctrines] by the sense of achievement you get from getting them to work at all. In other words... their fundamental design flaws are completely hidden by their superficial design flaws.†

ABSTRACT

Code in the particular form of software or so-called “algorithms” has presented a problem that has baffled, and embarrassed, patent law for as long as the problem has been acknowledged. Of course, prior to the time of wide-spread reprogrammable microprocessors, and before most people realized they were dealing with code, the problems were no less real, just hidden from view. The major computer software case now pending (CLS Bank Corp. v. Alice Corp.), coupled with the now unmistakable appearance of clearly biological code present a rare chance for a “do-over” in the law. Biological code is being used both for moving information traffic (by data storage, transmission and retrieval) and for changing the state of a machine, virtual machine or another coded construct, living organism, or biological carrier. These manifestations of code in a coded world may provide the impetus for finally getting the law’s response to code right.

I propose a functional definition of “code.” within a “coded world.” That is, let it be postulated, realistically, that “code” is a step or a series of steps for performing a function that can be represented by logical operators, wherein the function comprises (a) moving, storing, or modifying information, by (b) changing the state of a machine, a virtual machine, another coded construct, a living organism, or a biological carrier (and a “virtual machine” is an objectively reproducible coded mechanism for transforming an input to an output according to a rule). Let it be postulated that the “code world” is an embodied, switched, network, characterized by the presence of code that can cause an effect on other coded objects within or affected by the network. These definitions correspond to the reality of the phenomena.

† Douglas Adams, SO LONG AND THANKS FOR ALL THE FISH (and repeated, with variations, in other of his works). Interestingly enough, Adams was referring to the technology, not to the patent law that might protect it.
There are several reasons for preferring this definition of code. One of them is that code so defined not only accurately describes the objective phenomena that is occurring but also bears a working relationship to legally operative categories of patentable subject matter, and is capable of forensic proof. This definition unifies the legal approach to so-called computer software code, biologically engineered DNA and similar codes, and to yet other related phenomena. At the same time, it is able to distinguish problems that are functionally distinct. That is, it permits separate consideration of “code” on the one hand and “mental steps” or “business methods’ on the other. This, in turn, creates a regime in which different answers to each are possible. The categories of “code” and “mental steps/business methods” are de-coupled.

By avoiding attempts to categorize the patent-eligibility of code in terms of the degree of “abstractness” of the ideas, the extent of the “mental steps” involved, the presence of some sort of “mathematical formula” followed by some “nontrivial post solution activity” that ties it to a machine or transformation of some non-coded object, or by the “naturally occurring” or “living/non-living” nature of the object upon which the code operates, the new functional definition presents a practical, principled, workable rule of thumb that actually corresponds both to the reality of code and to the legal categories that have long characterized the analysis of patentable subject matter. It also makes economic sense and is efficient.

This neo-tech approach to code is not limited to patent eligibility. It presents opportunities for further work in related legal fields which are currently struggling to assimilate the important plurality of new cases that exhibit coded exceptionalism in a coded world that impacts the world of ordinary space. Among these are issues in cognate fields of intellectual property including copyright (the problems of ease of remix enabled by code, or the library in cyberspace made possible by code, or of code as a functional/useful article), and trademark (the problem of “invisible” or hidden trademark-containing triggers or magnets created by code). Perhaps more important than the coded problems in IP are the difficult problems raised by code in the analysis of first amendment speech (the problem, not merely of crying “fire” in a crowded theater, but of uttering a coded command that actually “burns” someone else’s data, or worse, by code that functions in a coded world like magic would have functioned in the ordinary world, as speech that is actually operative to change objective reality upon utterance or execution), fourth amendment search problems, privacy and encryption/decryption problems, problems of “hacking” an implanted bio-device, and the like. The coincidence of major pending cases that permit a reconsideration of the definitions, combined with the oncoming wave of neo-coded uses that will cause significant harm not readily resolved by current legal
regimes makes this an especially suitable time to rethink “code” and to get it more nearly right.

***

[NOTE: To conference organizers. This “abstract” as submitted is some 970 words. Please feel free to cut the last two paragraphs, or to cut the initial drop quote to make it more nearly fit, if it is too long as submitted]