

From: Jared Updike [e-mail redacted]  
Sent: Sunday, September 26, 2010 8:43 PM  
To: Bilski\_Guidance  
Cc: [e-mail redacted]  
Subject: Bilski decision and software patents -- just say no

26 September 2010

Dear USPTO,

Software patents hurt me, as an individual developer, and limit the thoughts I can think and the programs I can make or modify: the ways I can communicate and interact with others. As a matter of freedom, a single software patent can create an insurmountable, and unjustifiable, legal hurdle for myself and many would-be developers. The first amendment says that "Congress shall make no law ... abridging the freedom of speech." If certain programs, algorithms, or mathematical ideas are patented, and only corporations with large legal teams/budgets can build certain types of software, this places limits on what developers such as myself can think, say, or express, since software *is* mathematics. There is no software "not in the abstract".

[1]

Any interpretation of the Supreme Court's recent Bilski decision should reject all software patent application claims. Any other interpretation betrays a lack of understanding of mathematics and the theory of computation and damages the public, putting the interests of large players ahead of the users and smaller creators of software. It has been nearly 75 years since Alan Turing's paper was published [2] yet the USPTO has failed to acknowledge what all freshman Computer Science students have learned: software can be translated by machine to work on any machine, meaning no software is truly tied to hardware.

Are mathematical proofs patentable? Of course not. Why should algorithms be? The two are equivalent. [3] The USPTO can, and should, exclude software from patent eligibility on the grounds that software consists only of mathematics, which is not patentable, and the combination of such software with a general-purpose computer is obvious.

Sincerely,

Jared Updike.  
Sherman Oaks, CA  
Concerned Citizen and Software Developer

[1] see <http://www.groklaw.net/article.php?story=20100713173032257> and

<http://www.groklaw.net/article.php?story=20091111151305785> . For an explanation of basic Theory of Computation geared at lawyers, please read: <http://www.groklaw.net/article.php?story=20091111151305785>

[2] It is possible to invent a single machine which can be used to compute any computable sequence, that is the simple statement of Turing. To say software is bound to a particular machine is to ignore the ability to translate that software to any other machine and have it perform exactly the same. From Wikipedia: "This finding is now taken for granted, but at the time (1936) it was considered astonishing. The model of computation that Turing called his 'universal machine'—'U' for short—is considered by some (cf Davis (2000)) to have been the fundamental theoretical breakthrough that led to the notion of the stored program computer."

[3] [http://en.wikipedia.org/wiki/Curry%E2%80%93Howard\\_correspondence](http://en.wikipedia.org/wiki/Curry%E2%80%93Howard_correspondence)  
-- The Curry–Howard correspondence is the direct relationship between computer programs and proofs in constructive mathematics. Also known as Curry–Howard isomorphism, proofs-as-programs correspondence and formulae-as-types correspondence, it is a generalization of a syntactic analogy between systems of formal logic and computational calculi that was first discovered by the American mathematician Haskell Curry and logician William Alvin Howard.