Donald Knuth and Software Patents

http://swpat.ffii.org/players/knuth/index.en.html

Workgroup\\swpatag@ffii.org

2003-12-15

Donald Knuth, pioneer and cult figure of informatics (computer science), author of some definitive monumental classics such as "The Art of Programming", finds that software patents are built on some basic misunderstandings, similar to the misunderstandings of certain provincial american legislators in the 19th century or the medieval catholic church. Computer programs are as abstract as any algorithm can be, Knuth says.

Contents

- **Knuth to PTO**[1]

  I am told that the courts are trying to make a distinction between mathematical algorithms and nonmathematical algorithms. To a computer scientist, this makes no sense, because every algorithm is as mathematical as anything could be. An algorithm is an abstract concept unrelated to physical laws of the universe.

  Nor is it possible to distinguish between "numerical" and "nonnumerical" algorithms, as if numbers were somehow different from other kinds of precise information. All data are numbers, and all numbers are data. Mathematicians work much more with symbolic entities than with numbers.

  Therefore the idea of passing laws that say some kinds of algorithms belong to mathematics and some do not strikes me as absurd as the 19th century attempts of the Indiana legislature to pass a law that the ratio of a circle's circumference to its diameter is exactly 3, not approximately

[1] http://lpf.ai.mit.edu/Patents/knuth-to-pto.txt

3.1416. It's like the medieval church ruling that the sun revolves about the earth. Man-made laws can be significantly helpful but not when they contradict fundamental truths.

- **Donald Knuth on Software Patents in C'T Magazine 2002/02**[2]

  My personal opinion is that algorithms are like mathematics, i.e. inherently non-patentable. It worries me that most patents are about simple ideas that I would expect my students to develop them as part of their homework. Sometimes there are exceptions, e.g. something as refined as the inner point method of linear programming[3], where one can really talk about a significant discovery. Yet for me that is still mathematics.

  I come from a mathematical culture where we don't charge money from people who use our theorems. There is the notion that mathematics is discovered rather than invented. If something was already there, how you patent it?

- **Bernhard Reiter 2002-02-25: Knuth in c't zu Swpat**[4]

  see also the ensuing debate with patent attorney Springorum, who tries to relativise Knuth's statements by portraying him as a hippie who doesn't need to care about economic survival.

- **NZZ-Porträt von Donald Knuth**[5]

[2]http://lists.ffii.org/archive/mails/swpat/2002/Feb/0097.html
[3]http://swpat.ffii.org/papers/konno95/index.ja.html
[4]http://lists.ffii.org/archive/mails/swpat/2002/Feb/0092.html
[5]http://www-x.nzz.ch/folio/curr/articles/haffner.html

- **Werner Koch 2002-03-03: Knuths erstes Zitat zu Swpat**[6]

  When the patenting of algorithms started, Knuth was skeptical but not clearly opposed. In vol 3, 2nd printing, pp 318 of his Art of Programming, he wrote

  > Goetz's read-forward oscillating sort has the somewhat dubious distinction of being one of the first algorithms to be patented as an algorithm instead of as a physical device [U.S. Patent 3380029 (April 23, 1968)], unless successfully contested, this means it is illegal to use the algorithm in a program without permission of the patentee. Bencher's read-backwars oscillating sort technique was patented by IBM several years later. [Thus, we have reached the end of the era when the joy of discovering a new algorithm was satisfaction enough! Since programming is strongly analogous to the fabrication of a machine, and since computer programs are now worth money, patented algorithms are inevitable. Of course the specter of people keeping new techniques completetly secret is far worse than the public appearance of algorithms which are proprietary for a limited time.]

[6]http://lists.ffii.org/archive/mails/swpat/2002/Mar/0003.html

- **another article about knuth**[7]

  mirrors at http://russnelson.com/fea-knuth.pdf and http://www.cse.buffalo.edu/ ajay/fea-knuth.pdf

  Question: What is your thinking about software patents? There is a big discussion going on in Europe right now about whether software should be patentable.

  Knuth: I'm against patents on things that any student should be expected to discover. There have been an awful lot of software patents in the U.S. for ideas that are completely trivial, and that bothers me a lot. There is an organization that has worked for many years to make patents on all the remaining trivial ideas and then make these available to everyone. The way patenting had been going was threatening to make the software industry stand still. Algorithms are inherently mathematical things that should be as unpatentable as the value of PI. But for something nontrivial, something like the interior point method for linear programming, there's more justification for somebody getting a right to license the method for a short time, instead of keeping it a trade secret. That's the whole idea of patents; the word patent means "to make public".

  I was trained in the culture of mathematics, so I'm not used to charging people a penny every time they use a theorem I proved. But I charge somebody for the time I spend telling them which theorem to apply. It's okay to charge for services and customization and improvement, but don't make the algorithms themselves proprietary.

  There's an interesting issue, though. Could you possibly have a patent on a positive integer? It is not inconceivable that if we took a million of the greatest supercomputers today and set them going, they could compute a certain 300-digit constant that would solve any NP-hard problem by taking the GCD of this constant with an input number, or by some other funny combination. This integer would require massive amounts of computation time to find, and if you knew that integer, then you could do all kinds of useful things. Now, is that integer really discovered by man? Or is it something that is God given? When we start thinking of complexity issues, we have to change our viewpoint as to what is in nature and what is invented.

[7] http://www.ams.org/notices/200203/fea-knuth.pdf

- **KONNO Hiroshi: Karmarkar Tokkyo to Software – Sûgaku wa Tokkyo ni naru ka?**[8]

  > Another great mathematician and his struggle against patent inflation in Japan

[8]http://swpat.ffii.org/papers/konno95/index.ja.html