On the Software Patenting Controversy

(for in-classroom use only)

Dr. Marek A. Suchenek ©
Department of Computer Science
California State University Dominguez Hills

August 15, 2011

Version with highlights.

Abstract

This paper complements Section 4.5 Patents for Inventions in Software of Sara Baase's "A Gift of Fire" (4th ed.) [Baa08].

Contents

1 The current law

Article 1 Section 8 [8] of the U.S. Constitution (Powers of Congress) states:

> "The Congress shall have Power [...] To promote the Progress of Science and useful Arts, by securing for limited Times to Authors and Inventors the exclusive Right to their respective Writings and Discoveries [...]."

It gives the Constitutional foundation for the following Patent Law (Title 35 U.S.C. § 101, Inventions patentable):

> "Whoever invents or discovers any new and useful process, machine, manufacture, or composition of matter, or any new and useful improvement thereof, may obtain a patent therefor, subject to the conditions and requirements of this title."

where § 100 of that Law defines "process" as "process, art or method," including "a new use of a known process, machine, manufacture, composition of matter, or material."

Since a computer program (or an algorithm) clearly falls into category "process" so defined, one could argue that in light of the Patent Law software should be patentable. But the question of software patentability turned out much more complicated that the Section 8 [8] and § 101 seem to suggest and for the last few decades has been hotly debated and a subject of considerable controversy. Over the years, the courts weighted in on that question with their evolving, if not contradictory, interpretations of the noun "process" in the context of its patentability, inventing new tests and criteria that clarified some relevant issues but left some others murky or even made them more complicated than they were before.

2 The controversy

In the early days of proliferation of computer technology, the U.S. Patent and Trademark Office was routinely refusing patents on inventions with software components. This stance culminated in 1968 when the Office issued new guidelines in which computer programs were expressly declared as unpatentable[1]. Rejection decisions based on these guidelines were routinely challenged, and some of them were reversed by the Court of Customs and Patent Appeals. Although the Court, eventually, softened the then prevailing doctrine of software unpatentability, the U.S. Supreme Court ruled twice (in 1972 and 1978), albeit in a narrow context, against granting patents for specific inventions that involved computer programs.

Software-related patents became a really controversial issue after the U.S. Supreme Court decision in the 1981 case of *Diamond v. Diehr* opened a narrow possibility for patents related to software as long as they were parts of otherwise patentable inventions or processes. In that case, the Court ordered granting a patent to a method of curing rubber the only novelty of

[1]This reluctance to look into applications of software patents is understandable; just imagine every programmer trying to patent each useful (in his opinion) code that he invented but no one else, so far, did, and the overwhelming burden on part of U.S. Pattent and Trademark Office to examine them all.

which was a use of the computer to time the heating process. As a result, in the following years over 100,000[2] programs (algorithms, to be more precise) have been granted patents, including the RSA public-key cryptography algorithm in 1983 (see Fig. 1 for a drawing from the RSA patent description), Lempel–Ziv–Welch data compression algorithm in 1985, and

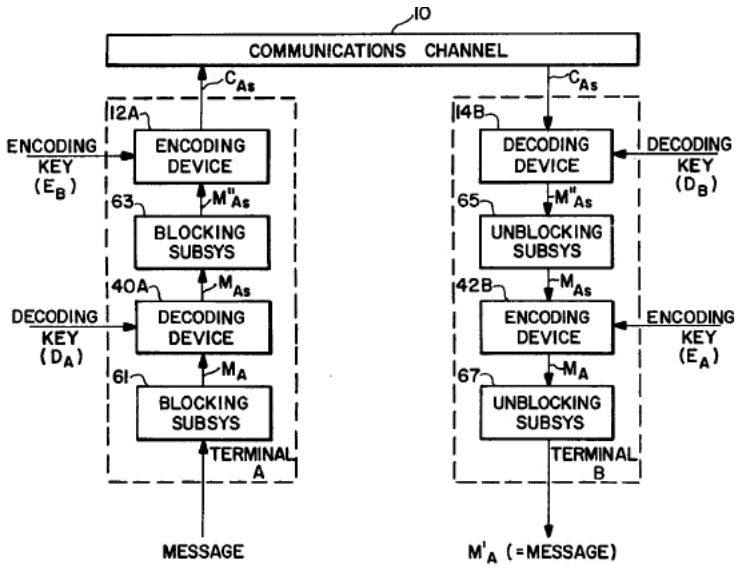

Figure 1: A drawing from the U.S. Patent 4405829 CRYPTOGRAPHIC COMMUNICATIONS SYSTEM AND METHOD.

Karmarkar's linear programming algorithm in 1988. In turn, a number of individuals (some of them prominent scientists) and organizations launched their crusades against software patents bringing up all sorts of arguments. For instance, Donald Knuth, in his undated (late 1980s) letter to Commissioner of Patents and Trademarks wrote:

> "[...] I strongly believe that the recent trend to patenting algorithms is of benefit only to a very small number of attorneys and inventors, while it is seriously harmful to the vast majority of people who want to do useful things with computers. [...] Please do what you can to reverse this alarming trend."

[2] An estimate.

(We will review some of the major arguments against software patentability in the sequel of this section.)

In 1998, the U.S. Court of Appeals for the Federal Circuit in the case *State Street Bank & Trust Co. v. Signature Financial Group* ruled that if software yielded "a useful, concrete, and tangible result" then it should be considered patentable. This ruling greatly facilitated patenting of computer software. But at the same time it invigorated those opposed to the idea. Various organizations and groups in the U.S. and elsewhere intensified their efforts to sway the public opinion against the concept of software patentability, at times resorting to quite persuasive language, like, for instance, in this 2006 headline: "[...] software patents rear their ugly head again" (a quote from nosoftwarepatents.com). As one could expect, they had their impact. And the emergence and proliferation of the so-called "patent trolls" - usually, law firms that specialized in buying a wide variety of patents solely for the purpose of subsequently suing manufacturers for patent infringements - fueled growing discontent with the legal post-1998 status quo.

Year 2007 marked the beginning of a reversal of the trend that made many software patents possible. The U.S. Court of Appeals for the Federal Circuit opinioned that if the invention consisted of a process that was entirely mental then it was unpatentable even if it was augmented with a modern electronics device, such as a computer. In what later became a highly publicized case, a patent examiner rejected the patent application for an algorithm that minimized risks in commodities trading "on the grounds that the invention is not implemented on a specific apparatus, merely manipulates an abstract idea, and solves a purely mathematical problem." That decision was challenged. In 2008, the U.S. Court of Appeals for the Federal Circuit in the case *In re Bilski* (the foregoing quotation comes from the text of that case) upheld that rejection, based on the so-called machine-or-transformation test for patentability, which in the Court's opinion the algorithm did not pass. Since most of computer programs were likely to fail that test, too, the Court's ruling was considered by many the last nail into coffin of software patents.

However, in 2010, the U.S. Supreme Court in the case of *Bilski v. Kappos* partially reversed the Federal Circuit's decision and rejected the machine-or-transformation test as the only criterion of patentability. Although the Supreme Court still ruled against granting patent for the said invention, it did not opinion against general patentability of any algorithm or a business process that failed the said test, but based its decision on the fact that a

patent for the invention in question would preempt an abstract idea[3], instead. These two rulings (in 2008 and 2010) resulted in an adoption by the Patent and Trademark Office a policy of refusing a patent to any computer program (referred to as a "method") that failed the machine-or-transformation test "unless there is a clear indication that the method is not drawn to an abstract idea," This has been a more stringent criterion than the "useful, concrete, and tangible result" test, thus making software patenting substantially more difficult than it was in the years 1998 - 2007.

Nevertheless, the Supreme Court was careful to not comment on software patentability, contenting itself with indication of the new difficulties that the Information Age raises for the patent law, and the necessity for reconciling the needs of inventors who seek the exclusive rights for their inventions and discoveries with the needs of the discovery process that may be affected by the pre-emption of procedures that are likely to be independently discovered by others. Wrote Justice Kennedy:

> "It is important to emphasize that the Court today is not commenting on the patentability of any particular invention, let alone holding that any of the above-mentioned technologies from the Information Age should or should not receive patent protection. [...] [T]he patent law faces a great challenge in striking the balance between protecting inventors and not granting monopolies over procedures that others would discover by independent, creative application of general principles."

The recently passed overhaul (*The America Invents Act, H.R. 1249*, signed into the law September 16, 2011) of the patent law did not directly address the patentability of software.

In the sequel of this section we will critically review some well known as well as some most common arguments against software patents. Also, we will quote and provide some that testify in favor of these patents.

---

[3] A standard criterion of non-patentability of an invention.

3 Constitutional and legal arguments

Stallman's argument ([Baa08], Section 4.6.2) with which he attempts to derive a lack of Constitutional protection of authors and inventors' compensation for their writings and discoveries and a lack of Constitutional basis for software patenting from the stated (in Section 8 [8] of the Constitution) power of Congress "to promote the Progress of Science and useful Arts" seems to contradict 9th Amendment to the Constitution. The mere fact that a right (in this case, the right to compensation for an intellectual property) has not been enumerated in the Constitution does not allow one to conclude that such a right does not exist or is unconstitutional. Such a conclusion would be particularly unwarranted in the context of Section 8 [8] that specifically mentions authors and inventors' "exclusive Right to their respective Writings and Discoveries." Although Stallman quotes the U.S. Supreme Court opinion in *Fox Film Corp. v. Doyal* that "United States has no interest" in this exclusive right "aside from the general benefits derived by the public", he fails to mention that said opinion was merely a premise for the Court's main judgment stating that "copyrights are not federal instrumentalities, and income derived from them is not immune from state taxation" and should be interpreted in that narrow context of income taxability, not as a summarily excuse to deny or to question constitutional protection of intellectual property rights of the creators of computer programs. Stallman's position seems to contradict the recent U.S. Supreme Court opinion in *Bilski v. Kappos*, in particular, Justice Kennedy's statement quoted above.

The recurring legal objection against patenting software that the Courts articulated while deciding appeals on patent rejections was a concern that a patent, if granted, would pre-empt a mathematical fact or an abstract idea[4], thus restricting others in their independent searches for new inventions and discoveries. This concern with potential pre-emption has been widely reverberated by the opponents of software patents. (Quite obviously, wherever there is a patent there is a pre-emption of some sort, or otherwise the patent would not serve its purpose, so this argument against pre-emption would - if accepted in its entire generality - lead to abolition of all patents.) What was conspicuously missing in arguments of this sort was a clear and commonly

[4]It has been a widely accepted legal doctrine that laws of nature, physical phenomena and abstract ideas are non-patentable.

accepted definition of "mathematical" and "abstract" - a lack with rather profound consequences as judges with no formal education in mathematics seemed ill-equipped to decide what was "mathematical" or "abstract" and what was not[5]. So, they struggled trying to sort this out. As a result, various tests were invented and applied, like - for instance - whether "a set of numbers is computed from a different set of numbers by merely performing a series of mathematical computations," which question if answered affirmatively would yield a conclusion that the algorithm in question is "mathematical". In 1989, the Office of the Solicitor of the Patent and Trademark Office released a legal analysis that attempted to define a concept of "mathematical algorithm" in the concept of patentability.

Aside from the fact that the specific objection against papentability of "mathematical" algorithm is hard to find in the code of the Patent Law[6] (the quoted analysis acknowledges that "[l]egislative history indicates that Congress contemplated that the subject matter provisions be given a broad construction and were intended to 'include anything under the sun that is made by man' "), the mentioned above efforts of courts and lawyers to differentiate between the "mathematical" and "non-mathematical" algorithms have been summarily criticized as inadequate or absurd, even by some opponents of software patenting. Writes Knuth in [Knu]:

> "I am told that the courts are trying to make a distinction between mathematical algorithms and non mathematical algorithms. To a computer scientist, this makes no sense, because every algorithm is as mathematical as anything could be. [...] 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.

[5]Not that it was the only term lacking a precise meaning; the adjective "obvious" gained some notoriety for being vague. For instance, what is obvious today might not have been then. After all, patents are supposed to explain things

[6]Courts did opinion in this matter, though; QUOTE *Mackay Radio & Telegraph Co. v. Radio Corp. of America*, 306 U.S. 86, 94 (1939). [Citations omitted]. The Supreme Court thus recognizes that mathematical algorithms are "the basic tools of scientific and technological work." *Benson*, 409 U.S. at 67, 175 USPQ at 674, and should not be the subject of exclusive rights, whereas technological application of scientific principles and mathematical algorithms furthers the constitutional purpose of promoting "the Progress of . . . Useful arts." U.S. Const. art. I, § 8. It is also recognized that mathematical algorithms may be the most precise way to described the invention.

> [...] 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 3.1416."

In some cases, algorithms have been thrown into the same category as mathematical formulas[7], like in *Diamond v. Diehr* that produced this rather controversial opinion: "[A]n algorithm, or mathematical formula, is like a law of nature, which cannot be the subject of a patent." They seemed to be a result of belief held by some judges that everything that can be precisely defined is somehow equivalent to a mathematical formula (its definition) and - therefore - unpatentable[8].

Not all courts subscribed to this idea. For instance, Justice Stone wrote for the U.S. Supreme Court in *Mackay Radio & Telegraph Co. v. Radio Corp. of America*:

> "While a scientific truth, or the mathematical expression of it, is not a patentable invention, a novel and useful structure created with the aid and knowledge of scientific truth may be."

In the light of these general and sometimes conflicting arguments it remains unclear what principle dictates that a drug formula, or a chemical process of making it, is a patentable matter in the Court's interpretation of the Patent Law but an algorithm ("mathematical" or not) is not.

---

[7] Try to find a mathematical formula equivalent to this Java program:

```
public static int f(int n)
{
      if (n <= 1) return 1;
      if (n%2 == 0) return (f(n/2) + 1);
            else return (f(3*n + 1) + 1);
};
```

for some seven decades now mathematicians and Computer Scientists weren't even able to figure out if this program halts for every integer n or not.

[8] For instance, this view was advocated by Ben Klemens in [Kle05]

4 Economic arguments

The distinction between "mathematical" or "abstract" and "non-mathematical" and "concrete" becomes particularly tricky in the Information Age. Is information "abstract" or not? Is it necessarily "mathematical"? (Some kind of it may well be.) Is processing an information by a computer program in order to derive from it, say, useful knowledge in the decision process (for instance, while controlling a technological process) a "mathematical[9] algorithm" in which "a set of numbers is computed from a different set of numbers by merely performing a series of mathematical computations"? Although it would be quite difficult to sell someone the Pythagorean Theorem, industrial and military spies are being paid fortunes for the information that they provide to their employers. Can something "abstract" or "mathematical" be traded for real money? It does not seem so.

Some economists (for instance, Kenneth Arrow in [Arr62]) argue that information[10] became a commodity that may or should be traded on a free market. They tend to see patents as market instruments that make such trading practically possible. In order to sell a commodity (say, for instance, a barrel of crude oil), the seller has to have a monopoly (the exclusive right, if you will) on a particular instance of that commodity, for otherwise prospective buyer will not have a reason to pay him for it. With information, the trading becomes tricky: unlike in the case of crude oil, once the seller has disclosed the information to the prospective buyer, he loses the monopoly on it so that the buyer has no incentive to pay for it. Moreover, the buyer may pass it for free to other prospective buyers just depriving the seller form the proceeds of future sales of the information in question.

This, considered by some controversial, view on information as a commodity sheds new light on the software patenting controversy.

The code of computer software certainly carries information (some would say *knowledge*): what decisions and what actions and in what order have to be made in order to accomplish the desired computational result. Copyright protects a particular expression of that information but not necessarily its contents, and certainly not the knowledge entailed by it. So, if information and knowledge (not just an expression thereof) can be traded then there

[9] And, therefore, non-patentable - according to the Supreme Court.

[10] The same can be said of knowledge.

seem to be no good reason why the software could not. In the existing legal system patent is the only practical means of protection that allows for trading information and knowledge carried by computer software. As much as it may lag behind the needs of the Information Age, it attempts to minimize the difficulty of creating a market for software the emergence of which, in turn, is likely to boost the supply of high quality software and not to impede it.

So, from the information-theoretic perspective, it does not make much economic sense to end software patents, contrary to what those opposed to patenting of software claim.

The same perspective provides means with which to disprove some other claims articulated against software patents. A commonly used anti-software patenting argument asserts that the inventor or creator of the software in question is not deprived of its use or other benefits just because other people are using it. But such an assertion is based on a fallacy that ignores the fact that the value of information depends not only on its contents but also on a lack of knowledge of it by prospective buyers[11]. In more information-theoretic terms, the value of information grows with its size measured as a minus logarithm of *a priori* probability that the information is true. So, as soon as it is known (and, perhaps, verified), the probability that it is true is 1 and its size becomes 0, as does its value.

Let's consider as an example a game of poker. Imagine yourself as a player. When you look at your poker hand, the information and the knowledge that you acquire this way has a very real monetary value to you. For instance, if you have a strong hand you may bet high with good probability of win. If, however, your hand is weak, you may consider folding, thus avoiding almost certain financial loss. You could not make a rational choice between the two actions if you did not know what your poker hand is.

Now, suppose one of players wants to see your hand, for his obvious benefit. He can call, which move requires on his part risking money that may end up in your pocket (if, for instance, you have the strongest hand). It is like if he offered you a patent[12] for your poker hand. You may consider

[11] This subjective dependence is referred to in economics literature as *asymmetry*.

[12] Copyright would not accomplish the same as simple re-arrangement of cards in your hand would become a different expression of the same information; but please remember that this example is nothing more than an informal illustration.

whether the value that he offers you this way is enough to show your hand. If it is not then you can rise a bet and keep playing, instead.

But what if one of the players demanded that you just show your poker hand for free without having any actual or potential benefits from such a move even though the rules of the game do not require you doing so? He may argue that by showing your hand to others you do not lose any use of the cards that you have in hand because you can still keep them without interference from other players and play with them any way you wish. If you accepted such an argument then I would love to play poker with you.

The above example illustrates fairly well the fallacy[13] that some of the staunch opponents of software patents[14] commit. It ignores what is called an asymmetry in information and information trading. And the poker example is a very good illustration of that fallacy. A software patent allows the inventor to materially benefit from disclosing the invention in public, and the general public benefits by learning, for a fee, from the disclosure how to make things or what decisions to make. Passing this information or knowledge on others for free and without reciprocity would cost you twice. Firstly, you would forfeit any revenues form future sales of the information and knowledge you own. Secondly, you will have to compete with those who took advantage of your generosity for customers, markets, raw materials, real property, and more. This may give them an upper hand that will let them outcompete you on the market. The second kind of loss may be particularly detrimental to you and your compatriots if the free transfer of said information is done internationally, as you have less chances of capturing any implied benefits from the increased productivity of your foreign competitors.

Another popular economic argument that is often brought up by the opponents of software patents begins with observation that they (the patents, that is) yield little, if any, economic benefit for the general public. In particular, it is argued, there is no convincing evidence that software patents spur any substantial innovation or progress in creation of new software. Some argue that the inventors of new software algorithms would invent them anyway,

[13] To see the fallacy in this superficially rational argument, apply it to money supply in order to conclude that printing more money and distributing it to others does not deprive their current holders of anything, or to fine art supply in order to conclude that making multiple copies of the same painting does not deprive an owner of the first one (the original) of anything.

[14] The same can be said of some advocates of free software, information, and knowledge.

whether rewarded with patents or not, and those who are not into inventing will, most likely, not invent anything of value even if lured by a prospect of handsome reward.

That may well be true. As a matter of fact, some economists (for instance, Gregory Clark in [Cla07]) collected substantial evidence showing that institutional means of protection of property (including intellectual property) hardly ever spurred any economic progress, at least not directly so. According to that view, the main function of these institutions is not to motivate any progress but to sustain one that is already in place. The framers of the Constitution might have hoped that protection of exclusive rights will "promote the Progress of Science and useful Arts", but - apparently - most that execution of this power of Congress has accomplished has been the nurturing of the progress that the authors and inventors are making just because they can. Once one realizes that, the argument against software patenting based on a lack of substantial evidence that software patents lead to more innovation loses its validity.

Protection of intellectual property in software may not spur innovation, but it is necessary for sustainability of that innovation. And since the copyright doesn't cover it well, what else, except patents, is there to protect discoveries of new algorithms?

The above observations, as well as evidence collected by some economists (e.g., by Arrow in [Arr62] and Clark in [Cla07] as mentioned above) seem to imply that a patent works more like a direct incentive to disclose the invention rather than to make it[15]. This explains why so many academics, accustomed and expected to publishing the results of their research without any extra compensation or other incentives[16], seem to understate the value of software patents.

On the long run, though, software patents, as well as other institutional and legal instruments of protection of intellectual and physical property rights, do contribute to economic and technological progress. In the absence thereof, many successful inventors, innovators, and entrepreneurs might have not decided to pursue careers in software industry, opting for other, more re-

[15] Clark reports that "[t]he establishment of [...] patent rights in northern Europe in the sixteenth century arose from the desire of countries to attract foreign artisans [...] [who] would not emigrate without legal guarantees that their knowledge would be protected."

[16] The requirement of publications for tenure is one of a few exception from that rule.

warding or profitable venues. Such and absence must result, eventually, in slowing the rate of innovation, as well as productivity, in software development. For instance, the likelihood of decent income (software patents are, however small, a part of it) has been one of the main factors that the parents of graduating high school students consider while influencing choices of their children's selection of a major. In the recent years, there has been a visible trend among the most talented students to pursue degrees in mathematics of finance and related areas - a choice that is often associated with prospects of high income and participation in profit sharing of stock trading and similar financial enterprises. If the expected rewards were equally appealing in computer industry, many of these highly talented students would have pursued degrees in computer-related subjects, instead. This almost certainly would spur a boom in software-related innovations. So, it is a fallacy to conclude that a lack of software patents does not hurt innovation[17] just because one believes that the patents are not a major motivation for invention.

An interesting insight into the question of appropriateness of ownership of information that may help to sort out some controversies pertaining to software patenting was offered by a Federal Judge Richard Posner (cf. [Pos78]). Prosner argued that in order to not discourage up-front investments in pursuit of information, useful information that was costly or difficult to acquire should allow for its ownership and protection, except that information that prevents such nuisances as fraud, deception, or misrepresentation (for instance, individual's criminal record and credit history) should - by default - be left in the public domain and, therefore, be presumed non-proprietary. Although Prosner was merely addressing certain privacy issues from economic perspective, his ideas appear strikingly congruent with provisions of the Patent Law that requires an element of difficulty or cost in a discovery of patentable invention and attempts to reconcile legitimate rights of the inventor with the needs of the society as a whole.

4.1 Global aspects

There are also some global aspects of the protection (or a lack thereof) of intellectual property rights that software patenting is an element of.

[17] Besides, if software patents were so detrimental to the progress in software creation as their critics seem to imply then the countries that historically allow little or no software patentability should be leading in development of software, which does not seem the case.

The recent proliferation of intellectual property, aided by emergence of inexpensive high-speed means of information and knowledge transfer and retrieval (whether lawful or not) that the Internet is and example of, often without compensation commensurate with the actual value of that property for the end user, coincides with the relatively new phenomenon of huge and growing trading imbalance in the U.S. measured as a share of the net export[18] in the GDP (see Fig. 2).

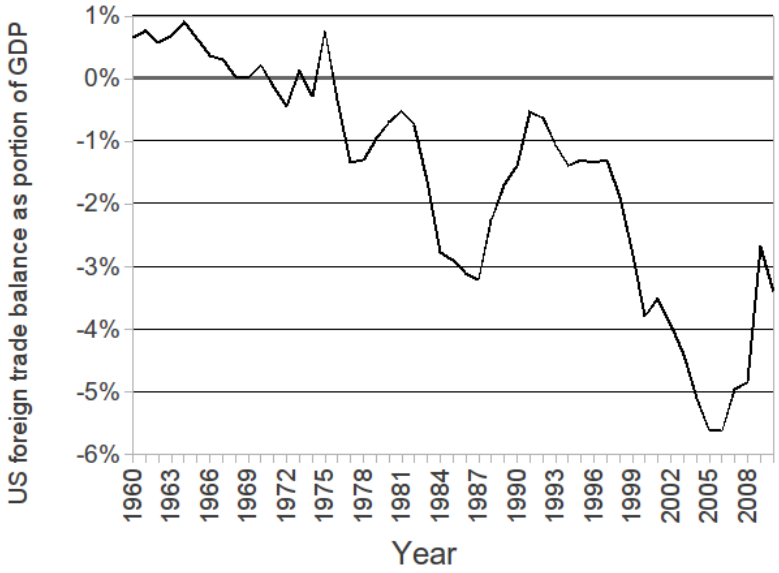

Figure 2: U.S. foreign trade balance measured as the portion of GDP. Calculated by the author from U.S. Census data (except for the value of GDP in 2010 that was based on an estimate of $14.7 trillion.)

Although direct revenues of export of intellectual property in general, and fees collected for licensing of patented software in particular, are but a fraction of the U.S. exports, it is difficult to not speculate that there exists a causality relationship between the proliferation of cheap or free intellectual property and the U.S. foreign trade deficit. Could it be that the falling revenues of transfer of intellectual property and growing world-wide expectation of free access to information and knowledge (including ones contained in

[18] The value of exports minus the value of imports.

software), commodities of the Information Age that that the U.S. has been historically the leading supplier of, have contributed to our current economic downturn? It is hard to see why they would not, particularly, if one agrees with some economists that information (knowledge) is the driving force of prosperity and creation of wealth, and that "innovations explains all modern growth" (Clark). A glance at a chart on Fig. 3 showing trends in prices of computer memory, the value of which is in part a derivative of the value of intellectual property needed to manufacture it, reveals its rather striking similarity to the U.S. trade deficit visualized on Fig. 2 page 15.

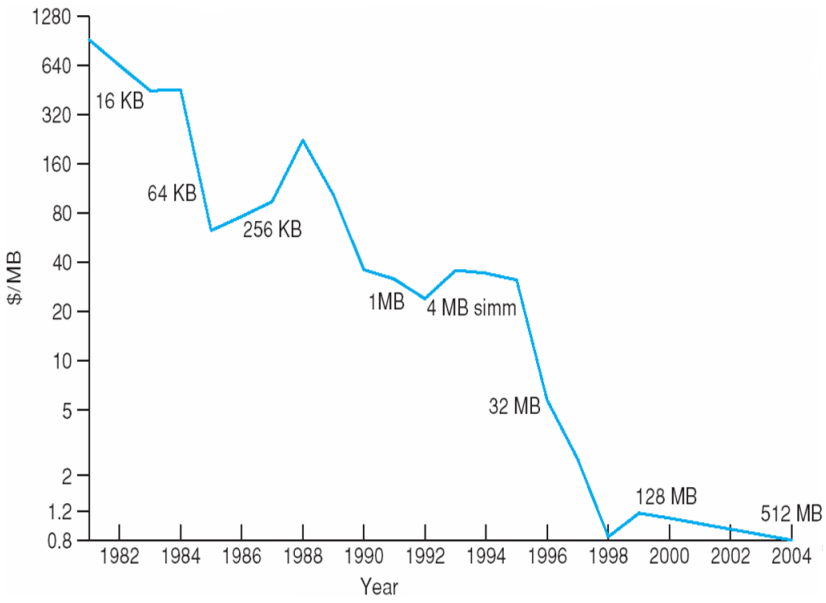

Figure 3: Price per Megabyte of DRAM, From 1981 to 2004. Source: Silberschatz, Galvin, *Operating System Concepts*, 8th edition (update), Addison-Wesley, Fig. 12.15 p. 542.)

According to some leading experts in global economy, not just the knowledge-based technology but an exclusive access to it is at the core of high living standards in the U.S. Write Donald Davis and David Weinstein in [DW05]:

"[O]ne can think of part of the income that U.S. natives enjoy as

> being based on their monopoly of access to the superior technology".

The analogy with poker game that was introduced earlier in this section (on page 11) illustrates how that monopoly of access works in the case of information technology.

The U.S. does not seem to be getting fair return on its huge investment in creation of knowledge, major benefactors of which are foreign customers. Even if one insists that software patents are a *passé* that need to be discontinued, now it certainly is a bad time to doing so as it would, most likely, add to the already huge[19] U.S. negative foreign trade account.

The lost revenues of sales of software and derived fees do not just hit workers. They also diminish collection of income taxes that are used to pay for social security, unemployment, and other public benefits that many take for granted.

5 Political arguments

It is worth remembering that for millennia economic arguments were used to justify slavery. So one may wish to scrutinize them from a political perspective as a precaution against undesirable consequences.

The main reason why the intellectual property is protected, it seems, is not to ensure that, say, Playboy Magazine makes huge profits on sales or that the authors of, say, popular songs become multi-millionaires. This hardly helps the economy, although the intellectual property in question contributes to the quality of people's lives.

Oddly enough, it is the software protection and not, say, motion picture protection, that is under the well orchestrated siege to which the existing legal system may give in. Meanwhile, no one pays any royalties for reciting mathematical theorems in public, and in some cases, the audience must be paid in order to make them listen to this kind of performances.

[19] Of the raw of magnitude of GDP.

The prices of computers and software have been falling rapidly. Part of this drop subtracted from the money that used to be paid to those instrumental to their creation, as entrepreneurs, researchers, designers, engineers, etc. Many of them lost their jobs due to layoffs[20], and many of those who did not, now have to work more and harder for lesser rewards. As if this weren't enough, public, now accustomed to cheap but powerful digital devices has developed a sense of entitlement to free software, a sentiment that has been fed and nurtured by various anti-software patenting organizations and prominent individuals.

But if the software were free than bulk of those who create it wold have to live of public generosity and charity of the philanthropists, competing with the recipients of welfare and other entitlees. Although many academics show strong preference towards federal/state sponsorship and privately funded grants as the primary (or the only) means of remuneration of the cognitive elite for their work in science and technology, they may find themselves in a difficult position (not that different from where other knowledge workers faced with prospects of layoffs are) these days when governmental budget shortfalls will likely result in deep cuts in spending on public universities and research institutions[21]. If the worst-case scenario materializes, they too will have to compete for means of subsistence with other benefactors of federal and state's generosity. One can hardly be an optimist in this respect as the governments of all levels have gain some notoriety for less than optimal or rational allocation of resources.

There seem to be a war on proprietary software nowadays[22], the mottos of which, *people need software* and *software creators make it just because they can*, came surprisingly close to one that for almost a century was tried by

[20] In 2010 alone, an est. 70,000 jobs were cut in U.S. electronics, computers and telecommunications industries.

[21] For instance, NASA space shuttle program has been recently discontinued in an attempt to narrow the federal budget gap. As if this weren't detrimental enough to national viability in the most advanced science and space technology, almost entire NASA was temporarily shut down on September 30, 2013, as a result of Federal Government fiscal insolvency, and its remarkably successful scientists and engineers were furloughed together with other "nonessential" governmental workers.

[22] The Internet is a prime, if somewhat one-sided, source of articles and posts that object, with varying degree of radicalism, the idea and practice of ownership of software and its patentability – one needs to dig really hard there to find serious papers that correctly present relevant and valid arguments in favor of software patents.

several nations to the detriment of their well-being: "to everyone according to his needs, from everyone according to his abilities"[23]. Not surprisingly, those opposed to the concept of intellectual property (or property at all) that they perceive as an obstacle to satisfy people's needs tend to lean against any forms of protection thereof, which makes software patents a natural target of their persistent criticism[24]. So their battle rhetoric shows a tendency to intensify when a major measure (like SOPA) to prevent software piracy is under consideration.

Some of the advocates of free information, free knowledge, and free software honestly disclose their aim: abandon all protection of intellectual property. This would be consistent with Marx who placed the right to profit from innovation with proletariat[25].

An example of such position may be found in a book "Against Intellectual Monopoly" by Michele Boldrin and David K. Levine [BL08], which among other qualities contains arguments that seem to go against Clark's of [Cla07]. The authors' main thesis is that intellectual property is in fact an "intellectual monopoly" that does not directly promote innovation, and that it hinders rather than helping free market competitiveness. From this they infer that intellectual property must have a diminishing effect on innovation and wealth that competition brings about. Their main argument seems to evolve around their assertion that "the only justification for intellectual property [protection] is that it increases – *de facto* and substantially – innovation and creation."

We have already noted that, as Clark has demonstrated in [Cla07], such an argument is invalid because the purpose of enforcement of intellectual property rights is protection of the existing and well functioning socio-economic system that delivers innovation rather than encouraging or intensifying it. The argument that the authors have devised is very strong. One can use it

[23] Marx's doctrine.

[24] Some would go to extra lengths while criticizing software patents (and other forms of protection of intellectual property) invoking free speech right guaranteed by First Amendment as *de facto* prohibition of restrictions on unauthorized copying. If one accepted such an argument as valid then it would render the "exclusive right to their respective Writings" clause of the Article I Section 2 of the U.S. Constitution meaningless. Hence, the fallacy of said argument.

[25] And the proletariat, as Clark in [Cla07] has discovered, captured the lion share of the benefits of the Industrial Revolution in England.

to dismiss, in logically same way, the need for criminal code based on otherwise, perhaps, correct observation that laws restrict freedom and no law can make a law abiding citizen out of a criminal. Such a dismissal would be fallacious as, according to some legal scholars[26], the main purpose of the law is, figuratively speaking, to protect the honest from the crooks rather than encourage honesty.

Boldrin and Levin provide very little and mostly anecdotical evidence that would support their claim that intellectual property is detrimental to progress in science and and, therefore, should be abolished. Implementation of their agenda, according to the argument brought up by some renown economists (for instance, Arrow in [Arr62]), will likely lead to collapse of the emerging free market for (non-free) information and knowledge trading, which in turn will, eventually, leave the bulk of knowledge workers at the mercy of public funding and philanthropy. Such a scenario may likely lead to decrease of living standards of many knowledge workers who may be forced (economically not physically) to become their own handymen, cleaners, cooks, plumbers, and mechanics, which eventuality will leave them with less time to pursue what they are presumably good at: creation of knowledge and innovation, even assuming that they are willing to continue creating it for free. The damaging long-term effect of such an arrangement to the innovation process and wealth that in the U.S. is predicated upon it is hard to overestimate. This fact seems to indicate that the arguments presented by Boldrin and Levin may be better characterized as political rather than economic[27] in their nature or intention.

Similar arguments against intellectual property in particular and capitalism[28] in general are being voiced by those firmly entrenched in the Marxian camp. For instance, Mike Palecek in "Capitalism Versus Science" [Pal09] seems to hold a very similar politically-economic line to Boldrin and Levin's when he writes:

> "We are constantly bombarded with the myth that capitalism drives innovation, technology, and scientific advancement. But in fact, the precise opposite is true. Capitalism is holding back

[26] With whom many political activists strongly disagree.

[27] Their cause, if come true, will inevitably trigger a cessation of free market of information and knowledge.

[28] For which property, physical or intellectual, is a cornerstone.

> every aspect of human development, and science and technology is no exception."

Those who assert such a position, which seems to posses all appearances of unproven nonsense, fail to explain why, under such circumstances, the flagship of anti-capitalist economy, the Soviet Union, fell behind the U.S. while heavily borrowing their "innovation, technology, and scientific advancement" from this country and did not become a beacon of innovation, instead.

6 Ethical arguments

There is a sentiment among some academics that all university faculty of a rank should have the same, modest, pay rate. The rationale for such an arrangement goes like this: the faculty who are capable of good teaching and scholarship do both because they can and not because of an expectation of extra compensation for academic excellence. The faculty who are not capable of any of these will not deliver no matter what monetary incentives are offered to them. Therefore, the said rationale goes, it does not make economic sense to pay more the former than the latter.

Besides the fact that such a system, where implemented, provides strong disincentive for talented and creative individuals to pursue academic careers (at least at the institutions that do not substantially reward academic excellence), there is something profoundly unethical in such an arrangement. Because the value of excellent academic work is higher than the value of mediocre academic work, never mind that the former requires substantially more effort to deliver than the latter, such a system would result in under appreciation of excellence and over appreciation of mediocrity, and lead to a pre-emption of some fruits of work (compensation) from those who deliver and awarding the pre-empted fruits of work who do not. And the reason for it is belief that substantial monetary awards for high volume and quality of academic productivity would, according to the advocates of equal pay for all within a rank, not yield measurable benefits for the institution[29]. It is like condoning injustice just because enforcing the law would not have sufficient

[29]Sentiment of this kind provides one more argument against moving the innovation process from free market to academia

economic benefits for the society as a whole. (Did any one say "utilitarianism"?)

A frequently used argument asserts that software should be free because people need *it* and *it*, the so-called *first copy*, is already there so *it* can be further copied and distributed at no cost (or at the cost of copying and distribution) to everyone that desires *it*. As we have seen, the fallacy involved in such an argument is the deriving a conclusion that it is socially unjust to charge a fee for a software from the fiction that copying of a computer program does not deprive its creator or owner of anything.

But putting aside the question of validity of logical reasoning that attempts to justify non-enforcement of intellectual property rights pertaining to software, be it in a form of copyright, be it in a form of patent, a fundamental question here is: does the inventor of computer algorithm have an obligation to share it for free with others? And if so then does this obligation begin at the moment he invented it, mentally, or when he put it in an implementable form that may be executed by a computer? Or does he have an up front (*continuous*, if you will) obligation to invent things that will benefit others to the extent that his talent, education, expertise, and willingness to work allow?

Let us assume, for the sake of an argument, that such an obligation does exist and is unconditional, that is, it is not contingent upon compensation for the inventor or a lack thereof. If the obligation begins after the work (usually hard and time-consuming) is done than such a system would have a tendency to discourage hard but presumably uncompensated work. If the obligation begins at the moment of mental invention that it would clearly violate one's right to privacy (as an obligation to disclose one's thoughts to public). If, however, it exists as an up front continuous obligation to invent, which many opponents of software patents seem to imply, then we approach surprisingly closely the "from every one according to his abilities" part of Marx's doctrine that wrecked the economies of several industrialized nations.

Even if we ignore all these objections, why is the intellectual property in general and computer software in particular different from commodities in the traditional sense of this word, as - for instance - crude oil?

Let's see. Usually, the only cost to the inventor for distributing free copies is a loss of revenue that would compensate him for his work that was needed to produce the first copy. But the same can be said of a small nation with

huge oil reserves. They will not be able to consume it all by themselves, so in the sense advocated by the opponents of software patents, it would not cost them anything (except, perhaps, for a lost of revenue from sales) to let others use portion of their crude oil for free.

But this is not going to happen anytime soon as the nations that control oil fields exercise their monopoly of access to these oil fields in order to get paid, and handsomely so[30], for something that they presumably haven't worked for. And those who question intellectual property rights and their protection do not seem to have problems with that, although it usually takes tremendous amount of work to create it. If they had their way then some (for instance, creators of software) would have to work hard for the benefit of others (for instance, those who control the access to oil fields) who earn their rewards with no or minimal work while refusing to return the favor. Such an arrangement raises some serious ethical objections. The U.S. is like a player who showed his poker hand but other players are not rushing to reciprocate. Once one realizes that, it somehow looks less surprising that the U.S. foreign trade deficit is huge and growing.

7 Other considerations

But the most fundamental question that needs to be answered before launching a crusade against software patents is whether intellectual property in general and computer algorithms in particular are overprotected. One thing seems clear: there is a widely spread anticipation (fear, if you will) that such an overprotection may happen. However, judging from various market indications and economic trends, that does not seem to be the case.

While the prices of software and products the functionality of which depends on it show visible decreasing tendency, the prices of energy, raw materials, and foods are steadily going up. And this disparity does not affect just the computer programs that fetch lower and lower prices that in some cases seem to converge to zero. Computers, digital devices, electronics, and many other categories of products with substantial intellectual property component

[30] They even have their own organization that pushes up crude oil prices on international markets.

that involves the state-of-the-art technology exhibit similar depreciation. For example, one cannot buy today even a barrel of crude oil for such a marvel of digital technology as a miniature 32 gigabyte USB removable memory device, despite the fact that the amount of hard and highly qualified work that is needed to invent, design, and setup manufacturing of the first copy of the latter is incompatible with the miniscule amount of work that it takes to trade the mineral right (the growing price of which is mainly responsible for the skyrocketing prices of crude oil) to the former.

The mentioned above disparity is a good illustration of how the intellectual wealth is quickly losing its value as a result of cheapening of intellectual property rights and the ease with which the intellectual property may be preempted without compensation or retribution. From the intellectual property budget perspective, less and less revenue, particularly when measured with prices of raw materials, is generated by a unit of the intellectual property content in the final product. In a market with unrestricted supply and restricted demand, and stagnant or very slowly growing productivity (think how hard it is to make progress in the most advanced areas of the cutting edge technology where we see people working at the limits of human intellectual productivity; a good example here is the area of Artificial Intelligence), it must reach a point where more and more work must be put into production of (the first copy of) intellectual property in order to recover a unit of revenue. Falling incomes of intellectual property creators (knowledge workers, if you will) and/or losses of IP-creating jobs, with all their negative fiscal consequences, like falling tax revenues, will characterize such a point if it is reached.

On the other hand, the proliferation of cheap intellectual property will, eventually, translate onto increased demand for raw materials and commodities (leading to higher prices thereof) paired with increased supply of knowledge workforce (thus suppressing wages in this group), further disadvantaging the already hard hit knowledge worker who now will have to compete with the beneficiaries of depreciation of the fruits of his work for such necessities as food, gas, and job.

Although it would be a stretch trying to imply that proliferation of software patents would stop the rapid growth of crud oil prices or shift the U.S. foreign trade balance from negative to positive, and we are not trying to suggest that, but certainly these is not the software patenting that should be a matter of concern of those sensitive to fairness and prosperity under the

current circumstances.

8 *Cui bono*: An inventor or his employer?

Hiring an inventor is a little like buying a license, but not a proprietorship, for a copyrighted or patented matter, except that the company by committing resources (honoraria, employment contracts, etc.) up front takes a risk of loss resulting from a subsequent lack of anticipated innovation or invention. But the risk factor does not necessarily counterweight the fact that it is the knowledge worker in question and not his employer who invented something. So, there is hardly a reason to *a priori* grant an employer summarily right to reproduce (or copy) the idea in the invention, unless it was properly sold by the inventor to his employer. This sentiment seems to be mirrored in the patent law that generally requires that patents for inventions be granted to actual inventors and not to third parties.

If one agrees that intellectual property rights deserve protection, the same seems true about the ability to generate knowledge possessed by the knowledge worker. The key factor in deciding how the value of the created knowledge is shared between the knowledge worker and his employer is the demonstrated (by means of a record of accomplishment and/or of education) ability to discover, innovate, and invent by the former that diminishes the risk of loss by the latter. If the knowledge worker does not show reasonable promise of generating knowledge then the employer's risk of loss resulting from a lack of revenue of his investment is high. In such a case, one can argue that the employer should own the intellectual property created in the course of employment by the knowledge worker. If, however, the knowledge worker is demonstrably able to create knowledge then the employer's risk is low and the said intellectual property belongs to the knowledge worker[31], who can sell it to the employer for additional compensation.

If an investor hires a gambler to bet at random at investor's cost then the win, if any, should belong to the investor (as does the risk) and the

[31] This realization may explain the origins of Marx's overly general and, therefore, fallacious doctrine that what he called a *surplus value* belongs to the worker (but *not* to knowledge worker) regardless of his contributions, or a lack thereof, to the knowledge component in the end product and employer's risk of loss.

gambler may be entitled just to an hourly wage for the time he spent playing on investor's behalf. But if the gambler possesses demonstrated capability of winning then he, perhaps in addition to wage, should have a share in the win. For instance, if it is known that the gambler always wins then the investor never takes a risk of loss, and then there is no good reason for awarding him any payoff above, say, a fixed interest on his investment[32].

9 What's in the future?

Gregory Clark in his seminal book "A Farewell to Alms: A Brief Economic History of the World" [Cla07] provided an insightful analysis of the Industrial Revolution in England around 1800. An ample evidence that Clark has collected (and used in his book in support of his theses) makes the surprising conclusions he draws a matter of fact. Since there are some striking similarities between the Industrial Revolution then and the Information Revolution now, one may wish to use Clark's work in an effort to gain a better perspective on the software patenting controversy.

Both revolutions were predicated upon unprecedented, large scale innovation, the driving force of which during the Industrial Revolution was the textile industry, and during the Information Revolution is the software industry. Yet during both revolutions, these were *not* the innovators who were the main beneficiaries[33] "of the social rewards their enterprise wrought." According to Clark, "the textile innovators of the Industrial Revolution, even those who were successful and are now famous, typically earned small returns," and the Information Revolution does not seem to depart from that pattern. In the recent years, computer industry laid-off thousands of software engineers and similar professionals due to various cost-cutting efforts that the sharply falling sales prices brought about. Yet the massive innovation in both periods continued, thus supporting the thesis that these were not the spoils and the rewards but the ability and willingness of the innovators that spurred it.

[32] In other words, in such a case all the gambler needs is a bank to borrow money from and not an investor.

[33] Writes Clark: "Wage earners and foreign customers, not entrepreneurs, were the overwhelming beneficiaries of innovation."

Clark found out that patent protections of innovations that made the Industrial Revolution were weak and that many textile innovators, including some famous ones, died in poverty. It is up to us to chose whether we want that history to repeat itself.

References

[Arr62] Kenneth Arrow. *Economic Welfare and the Allocation of Resources for Invention*, pages 609 – 626. in The Rate and Direction of Inventive Activity: Economic and Social Factors. UMI, 1962.

[Baa08] Sara Baase. *A Gift of Fire: Social, Legal, and Ethical Issues for Computing and the Internet*. Pearson/Prentice Hall, 2008.

[BL08] Michele Boldrin and David K. Levine. *Against Intellectual Monopoly*. Cambridge University Press, 2008.

[Cla07] Gregory Clark. *A Farewell to Alms: A Brief Economic History of the World*. Princeton University Press, 2007.

[DW05] Donald R. Davis and David E. Weinstein. United states technological superiority and the losses from migration. *Backgrounder*, February 2005. Center for Immigration Studies.

[Kle05] Ben Klemens. *Math You Can't Use: Patents, Copyright, and Software*. Brookings Institution Press, 2005.

[Knu] Donald Knuth. Letter to the patent office: Commissioner of patents and trademarks.

[Pal09] Mike Palecek. Capitalism versus science. *In Defence of Marxism*, August 2009. http://www.marxist.com/.

[Pos78] Richard Posner. An economic theory of privacy. *Regulation*, pages 19–26, May/June 1978.