

COMPUTERS AND THE PATENT SYSTEM: THE PROBLEM OF THE SECOND STEP

Orin S. Kerr*

The proper scope of patent protection for computerized algorithms is one of the most controversial questions in modern patent law. Everyone has an opinion. To some, broad patent protection for computer algorithms properly updates the patent system for the Internet age.¹ To others, it signals a dramatic and unwarranted expansion of the scope of patentability.² The appropriate scope of protection for computerized algorithms has proved to be a remarkably difficult question. After almost forty years of debate and case law development,³ we seem no closer to a

*Associate Professor, George Washington University Law School; B.S.E., mechanical engineering, Princeton University; M.S., mechanical engineering, Stanford University; J.D., Harvard Law School. This essay is a revised version of remarks I delivered on April 12, 2001, at a symposium sponsored by the *Rutgers Computer and Technology Law Journal* at Rutgers Law School in Newark. At the time of the symposium, I was a Trial Attorney with the Computer Crime and Intellectual Property Section of the United States Department of Justice. However, the opinions that appear in this essay are mine alone, and do not reflect the position of the United States Department of Justice. I would like to thank John Duffy, Scott Kieff, and Nathan Judish for their helpful comments on a prior draft, and the editors of the *Rutgers Computer and Technology Law Journal* for kindly inviting me to speak at the symposium.

1. See, e.g., Gregory J. Maier & Robert C. Mattson, *State Street Bank in the Context of the Software Patent Saga*, 8 GEO. MASON L. REV. 307 (1999); Chad King, Note, *Abort, Retry, Fail: Protection for Software-Related Inventions in the Wake of State Street Bank & Trust Co. v. Signature Financial Group*, 85 CORNELL L. REV. 1118 (2000).

2. See, e.g., John Thomas, *The Patenting of the Liberal Professions*, 40 B.C. L. REV. 1139 (1999); Rochelle Cooper Dreyfuss, Essay, *Are Business Method Patents Bad for Business?*, 16 SANTA CLARA COMPUTER & HIGH TECH. L.J. 263 (2000).

3. See, e.g., Comment, *The Patentability of Computer Programs*, 38 N.Y.U. L. REV. 891 (1963); *In re Prater*, 415 F.2d 1378 modified on rehearing, 415 F.2d

consensus than when we started.

I will not attempt to resolve this thorny question here. Instead, I want to discuss why the scope of patent protection for computerized algorithms has proved to be so difficult. Why have computers created such a headache for the patent system? My argument is that the invention of the general purpose computer has baffled the patent system by breaking down the traditional one-step analytical framework of new inventions into two analytical steps. Unlike traditional machines and processes, general purpose computers divide the brains of the operation (the algorithm) from the brawn (the hardware). Patent law is ill-equipped to respond to this bifurcation. Instead, patent law's one-step conceptual framework forces us to confront an all-or-nothing choice between two unsatisfying alternatives: either we can grant computerized algorithms too much protection, or too little. Either every algorithm is patentable subject matter, or none are.

More broadly, the computer's bifurcation of brains and brawn restructures the relationship among invention, action, and value that the patent system presumes. The patent laws were designed to encourage inventors to discover and share knowledge about how we can manipulate the physical world.⁴ Invent a new mousetrap, and the government will give you a patent. That made a lot of sense back when value and physicality were closely linked. Today, however, general purpose computers have decoupled value and physicality: value now derives more from the software brains than the hardware brawn. So, which do we choose? Physicality or value? Like a three dimensional visitor to a two dimensional world,⁵ we are forced to squeeze the new reality into preexisting legal rules, with unexpected results. Either we offer protection to all algorithms, or none. Our difficulty in finding the proper scope of protection for computerized algorithms reflects our grappling

1393 (C.C.P.A. 1969).

4. See Orin S. Kerr, *Are We Overprotecting Code? Thoughts on First-Generation Internet Law*, 57 WASH. & LEE. L. REV. 1287, 1295 (2000) ("What distinguished patentable inventions from merely interesting ideas was that the former announced a new way that the natural world of realspace could be manipulated to reach a practical result.").

5. See EDWIN A. ABBOTT, *FLATLAND* (Little, Brown, and Company 1937) (1884).

with these unsatisfying options.

I. THE ONE-STEP FRAMEWORK OF CLASSICAL PATENT LAW

Imagine the world before computers. In those days, the scope of patent protection was conceptually straightforward. For the most part, you could obtain a patent for two types of inventions: machines and processes.⁶ Both machines and processes followed a simple analytical model. You started with some kind of input; you ran the input through the machine or process; and you were left with some kind of output.⁷ This one-step, input-to-output conversion characterized every machine and process, no matter how complex.

Consider Eli Whitney's cotton gin, patented in 1794.⁸ The cotton gin solved a major problem that had plagued cotton growers worldwide: freshly picked cotton contained seeds and other debris that had to be hand-picked by laborers, who could only yield a few pounds of cleaned cotton per day.⁹ Whitney's cotton gin was a mechanical cotton-cleaning machine that separated the cotton lint from the seeds and debris ten to a hundred times faster than a person could.¹⁰ The cotton gin followed a simple analytical model: you began with the uncleaned cotton, ran it through the machine, and were left with cleaned cotton in one bin and debris and seeds in another.¹¹ You started with input, ran it through the machine, and were left with output.

Thomas Edison's lightbulb provides another example. Edison's lightbulb was a revolutionary machine that cheaply and reliably converted electrical energy into light. To operate the lightbulb, you inputted electrical energy into the machine, the energy excited a

6. See 35 U.S.C. § 101 (1994). Patents can also be obtained for manufactures and compositions of matter. See *id.* However, for the purpose of this article, I will focus on machines and processes.

7. See *Gottschalk v. Benson*, 409 U.S. 63, 70 (1972).

8. Whitney's invention and the history of his patent application are described at www.nara.gov/education/cc/whitney.html. (last modified Mar. 12, 2001) (on file with the Rutgers Computer and Technology Law Journal).

9. See LEON H. AMDUR, PATENT FUNDAMENTALS 14 (1959). This grueling work was usually performed by slaves. *Id.*

10. See *id.* at 15.

11. *Id.* at 14-15.

filament that glowed brightly, and you enjoyed the illuminating light. Like Whitney's cotton gin, Edison's machine acted as a way of converting input into output; in this case, electricity into light.

Patentable processes followed a similar analytical path. Whereas machines were physical devices that manipulated input into output, processes explained a series of steps that could manipulate input into output.¹² Consider Charles Goodyear's process for treating (or "vulcanizing") rubber, patented in 1844.¹³ Goodyear discovered that when he treated raw rubber with sulphur and subjected it to high temperatures and pressures for specific periods of time, the rubber could be transformed from its raw state into a durable, useful industrial commodity.¹⁴ Goodyear did not invent a specific rubber-treating machine; instead, he obtained his patent for the process of treating raw rubber.¹⁵ The analytical framework for his invention was simple: you began with the input of raw rubber, manipulated it through the steps Goodyear had developed, and were left with vulcanized rubber as output.

I have focused on the one-step framework because I think it is the key to understanding how the patent system traditionally approaches the scope of patentability. Although the Patent Act formally describes the scope of patentability in terms of "machine[s]" and "process[es]," the actual administration of the patent system has permitted patents to be issued for any means of "transform[ing] and reduc[ing] an article to a different state or thing."¹⁶ The scope of patentability has been defined by the one-

12. As the Supreme Court stated in *Cochrane v. Deener*:

A process is a mode of treatment of certain materials to produce a given result. It is an act, or a series of acts, performed upon the subject-matter to be transformed and reduced to a different state or thing. If new and useful, it is just as patentable as is a piece of machinery.

94 U.S. 780, 788 (1876).

13. *Tilghman v. Proctor*, 102 U.S. 707, 722 (1881). See also *AMDUR*, *supra* note 6, at 16-18. Goodyear obtained patent number 3,633 for this invention. See *AMDUR*, *supra* note 9, at 18.

14. *AMDUR*, *supra* note 9, at 18.

15. See *id.* at 17-18.

16. *Gottschalk v. Benson*, 409 U.S. 63, 70 (1972) (internal quotations omitted). Of course, the requirement of patentable subject matter is only one of several prerequisites that must be satisfied before a patent will be issued. See Orin S. Kerr, *Rethinking Patent Law in the Administrative State*, 42 WM. & MARY L.

step input-to-output conversion: any way of converting some kind of real-world input into some kind of real-world output can qualify. If the means is a physical device, the invention is a “machine”; if it’s not, it’s a “process.” The statutory right to exclude granted by a patent has been available to any device or procedure that can convert input into output and meets the remaining requirements of patentability such as novelty, non-obviousness, and utility.¹⁸

II. THE TWO-STEP FRAMEWORK OF COMPUTERS

Now fast-forward to the present. Computers have changed everything. Unlike inventions such as the cotton gin and the lightbulb, computers are general purpose machines that can be programmed to do just about anything. My desktop computer can be typewriter, a numerical calculator, or a pinball machine depending on whether I am running Wordperfect, a spreadsheet, or a game file. What my computer is depends on what kind of software I am running. The hardware itself simply provides a platform for an operating system, which itself provides a platform for software applications. In the place of a single “machine” or “process,” general purpose computers have divided the brains of the operation from the brawn. On one hand, you have the physical hardware that executes the instructions, and on the other hand, you have the software that tells the hardware what to do.¹⁹

A funny thing happens when we try to approach general-purpose computers from the one-step framework of classical patent law. It doesn’t quite fit, because the single step has been divided into two. In the computer world, we take the user’s input, feed it into the software application running on the hardware, and then the

REV. 127, 136-37 (2000) (discussing the Patent Acts’ statutory requirements for patentability).

18. Admittedly, the one-step dynamic is more apparent with some inventions than with others. For example, some inventions are designed to prevent physical damage to property, and they “work” when they protect the property from harm. Although it may seem odd to view such inventions as means of converting input to output, I think the analytical framework still applies. Consider the case of a bullet-proof vest. The vest is designed to absorb a bullet and distribute its momentum. While it may seem to be simply a “thing,” it is also a means of converting input into output: in this case, a moving bullet into a stopped bullet.

19. I recognize that this explanation oversimplifies how computers work. However, I think the added clarity outweighs the loss of technical accuracy.

software directs the hardware to perform the necessary steps to create the output. Consider what I did when I wrote this article on my computer. I loaded Windows 95 and WordPerfect onto the computer, and then typed in the article using the keyboard. The software directed the hardware to process the key strokes, resulting in the output of a law review article. The old “machine” has been broken down into two basic parts: the software and the hardware, the brains and the brawn.

This bifurcation of roles poses a difficult problem for the patent system. If the role of a patentable “machine” is now occupied by a combination of hardware and software, what within this new regime should receive patent protection? Should only the hardware be patentable? Or a specific combination of hardware and software? Or perhaps both the hardware and the software should themselves be individually patentable? More abstractly, when technology allows us to divide a machine into its logical component parts, dividing a one-step process into two, which of the parts should be entitled to patent protection? Nothing in patent law offers an easy answer to this question; we’re just not used to this new distribution of functional roles. What we *are* used to is fitting the scope of patentability into the one-step framework. Start with input, end with output. How can general purpose computers fit?

III. ALL OR NOTHING: GENERAL PURPOSE COMPUTERS AND THE ONE-STEP FRAMEWORK

When we try to fit general purpose computers into the one-step framework, we face a choice between only two coherent approaches: either we extend broad patent protection to computerized algorithms, or else, we extend almost no protection to them.

Let’s start with the first case, computers as machines. Here, the scope of patentability depends entirely on how we define the machine. Is the software algorithm itself part of the machine, or is it simply part of the input? If we view the software as part of the machine, then the brain of the machine merges into the brawn. The computer acts as a black box, and a computer running one program is a distinct “machine” from the same computer running a different program. This approach creates strikingly broad patent protection

for computerized algorithms. By obtaining a patent on a machine that executes a particular algorithm, an inventor earns the right to exclude others from running that algorithm on a computer for the duration of the patent. Because the physical hardware is almost always interchangeable, the patent effectively bestows a right to exclude others from executing the computerized algorithm. And any distinct algorithm will do. If the algorithm manipulates input (whether from the keyboard, mouse, or software file) and produces some kind of output, it can be a distinct “machine” that falls within the scope of patentability.

There is, however, a second way to fit computers into a one-step framework of patentable machines, and it leads to very different results. Instead of treating the algorithm as part of the machine, we might decide to treat the algorithm as simply part of the input to the machine. From this perspective, a computer user inputs software and any additional data into the hardware machine and enjoys the output directed by the software. The machine is defined as only the physical hardware, and it is the same machine regardless of what software the computer happens to run. This approach results in effectively zero protection for computerized algorithms. The algorithm ceases to be part of the machine and the patent system can no longer bestow an exclusive right to it. Instead, the algorithm is merely input to be fed into the hardware machine. Depending on which approach we take, either all algorithms potentially fall within the scope of patentable subject matter, or none do.

A similar dichotomy emerges when we look at patentable processes instead of machines. We confront the following question: does a computerized algorithm manipulate data enough that the algorithm’s output is “transformed and reduced to a different state or thing?”²⁰ In other words, does the algorithm alone constitute a patentable process? Imagine that I develop an algorithm that performs a series of mathematical operations on a set of numbers, and produces a second set of numbers. Can I fit that into the one-step framework of patent law by labeling the first set

20 *Diamond v. Diehr*, 450 U.S. 175, 183 (1981) (quoting *Cochrane v. Deener*, 94 U.S. 780, 787-88 (1877)).

of numbers the “input,” the algorithm the “process,” and the second set of numbers the “output”? If the answer is yes, then any computerized algorithm can be a patentable process. Every series of instructions transforms input to output. If the answer is no, however, then no computerized algorithm can be a patentable process. The algorithm can only direct the hardware to perform a series of calculations, no matter how complex, and cannot actually “transform” input into output in the way characteristic of patentable processes. Again, either all algorithms can fall within the scope of patentable subject matter, or none can. It’s all or nothing.

But algorithms running on general purpose computers are something new: part input, part machine. They both integrate and define the hardware and exist independently from it. As a result, any attempt to shoehorn the algorithm of a general purpose computer into the traditional one-step framework of patent law leads to the conclusion that either everything can be protected, or nothing can be. The one-step framework offers no conceptual middle ground that might extend patent protection in some circumstances, but not in others.

IV. WHY ALL-OR-NOTHING PROVES UNSATISFYING

I think that neither of these approaches is particularly satisfying. In this section, I will explain why.

Consider the prospect of providing broad patent protection to computerized algorithms. Such a stance is deeply inconsistent with the historical function of the patent laws. Traditionally, patent laws have existed because the physical world is difficult to master: while it’s easy for an inventor come up with a new idea in his head, it’s usually hard to implement the idea in the real world. For example, the idea of a lightbulb was obvious to everyone in the Nineteenth Century, but it took Thomas Edison thousands of hours of experimentation before his actually worked.²¹ The patent laws respond to this by offering incentives for the discovery and disclosure of new ways of manipulating the world. Invent a new machine or process that does something useful with the world, and

21. [Http://homestead.juno.com/pdeisch/files/bulb.htm](http://homestead.juno.com/pdeisch/files/bulb.htm) (last visited Oct. 18, 2001).

you can obtain the *quid pro quo* of a patent.²² In contrast, come up with an abstract idea, such as a new theory of democracy or the ending of a movie, and the patent system gives you nothing.²³ This dividing line makes functional sense. If discovering how to manipulate the physical world is hard, but coming up with an idea is easy, it makes sense to offer an incentive for the former but not the latter.

Offering broad patent protections for algorithms contradicts this historical distinction. Algorithms are sets of logical instructions that do not hinge on interaction with the real world: they embody ideas, theories, and insights, but not the ideas, theories, and insights of the physical world. They are mere ideas, albeit mere ideas programmed onto a computer. Once we leap the conceptual divide and start granting protection for new ideas that run on computers, it's hard to see where we can stop. Most ideas can be programmed on a computer, which means that pretty soon we have to offer protection for new ideas, period. Come up with a good idea- about anything- and you will be able to obtain a right to exclude others from using your idea for twenty years. Granted, this expansion of the scope of patent protection may seem desirable at first. Expanding the scope of patent protection to include X will always produce an incentive to create more X, at least in the short term.²⁴ And who doesn't want more good ideas? The downside, however, is the long term consequence of privatizing ideas by creating property rights in their use. If we expand patent protection to mere ideas, will the long-term cost to the public domain be worth the short term gains? The historical answer, at least, has been "no."

At the same time, I find the idea of offering no protection for algorithms to be equally unsatisfying. I have two reasons. First, I find it hard to draw a sharp line between a general purpose computer running particular software, and a machine that contains a computer that has been "hard-wired" to run a particular program. Today, many everyday devices contain computers of one sort or

22. See Kerr, *supra* note 16, at 135-38 (discussing offer and acceptance of patent terms based on congressional incentive to encourage discovery).

23. See Kerr, *supra* note 4, at 1295-96.

24. If they gave away patents for wearing clown suits to work, you would see a lot of people coming to work in clown suits.

another, and they are integrated seamlessly into the product. For example, your refrigerator may have a computer inside of it, as may your microwave oven. It seems obvious that the patentability of such machines should not hinge upon whether they happen to contain computers, as opposed to merely mechanical or electric mechanisms.²⁵ But if we accept that, then it's hard to see where we can draw the line to separate these machines from general purpose computers running particular software. The doctrine simply doesn't give us a ready hook upon which to distinguish brains from brawn.

The second problem with not extending patent protection to algorithms is that they seem too integral to modern technologies and technological progress to receive no patent protection whatsoever. Algorithms are not merely inputs; they are the souls of modern machines. From a rational perspective, I admit, this is a fairly weak, almost extra-legal argument. Its flaw is that the patent system has not been tasked with the job of identifying whatever appears economically valuable in our society, and then claiming it as its own. If new technologies do not hinge on interaction with the real world, we might say, then so be it; let them develop outside of the patent system. At a gut level, however, it seems right that the patent system should matter here. For over two hundred years, the patent system has regulated rights in new inventions. Why stop now?

V. THE FEDERAL CIRCUIT AND THE CHOICE OF GREATER PROTECTION

Let's now turn from theory to law. How have the courts responded to these problems? How have the courts, and in particular the Federal Circuit, applied the one-step framework of classical patent law to general purpose computers? The answer is that the Federal Circuit has recently embraced the choice of broad patent protection for computer algorithms, with all of its implications. The Federal Circuit has applied the one-step framework by including the algorithm of a general purpose computer as part of the "machine," and by holding that the

25. *See Diehr*, 450 U.S. at 187-88.

algorithm alone can constitute a “process.”²⁶

In 1998, the Federal Circuit announced its landmark decision on how the one-step framework for machines applies to general purpose computers.²⁷ *State Street* involved a patent for a machine described as “a data processing system for managing a financial services configuration” relating to mutual funds that comprised a “computer processor,” “storage means,” and means for processing data to calculate certain financial data such as income, expenses, gain and loss to a particular portfolio.²⁸ The patent granted its owner an exclusive right to use a computer to execute a series of calculations that represented a particular configuration for pooling mutual funds.²⁹ The real value of the invention derived from the fact that the particular way of pooling funds was easy for fund managers to administer and carried certain tax benefits under existing tax laws.³⁰ Instead of trying to patent the way of pooling funds, however, the inventor attempted to claim a machine: a general purpose computer programmed to execute the inventor’s way of pooling funds.

The Federal Circuit held that this invention did in fact claim a patentable machine, and that any other conclusion would improperly “read limitations . . . on the subject matter that may be patented.”³¹ The fact that the value and novelty of the claimed machine derived from its presumably unpatentable algorithm was irrelevant, the court stated, because “[t]he dispositive inquiry is whether the claim *as a whole* is directed to statutory subject matter.”³² When read “as a whole,” the claimed invention was simply a black box - a machine that performed a valuable function by managing mutual funds.³³ Under the one-step framework, the claimed invention was identical to a mechanical or electrical device that was hard-wired to perform the same task. The hardware and

26. See *infra* notes 29-43 and accompanying text.

27. See *State Street Bank & Trust Co. v. Signature Fin. Group, Inc.*, 149 F.3d 1368 (Fed. Cir. 1998).

28. *Id.* at 1371-72 (describing an independent claim and written description).

29. See *id.* at 1371.

30. See *id.*

31. *Id.* at 1373.

32. *Id.* at 1374 n.6 (citing *Diehr*, 450 U.S. at 187) (emphasis added).

33. See *id.*

software merged into a single machine that fell within the scope of patentability. As I noted above, this appears perfectly logical within the one-step framework, but also results in a dramatic expansion of the scope of the patent laws.

The Federal Circuit applied a similar one-step framework to computerized processes in *AT&T v. Excel Communications*,³⁴ within a year of *State Street*. The inventor in *Excel* claimed a technique for simplifying the billing of long distance telephone customers based on their long distance providers. Rather than maintain a complex set of bills to determine the rates for different customers, long-distance providers could keep a simple record of the long-distance carriers assigned to the parties to a particular call and then send that information to other long-distance carriers that carried the call.³⁵ Carriers could then look at the information, perform a few quick calculations, and determine how to bill a particular call.³⁶ The inventor applied for and obtained a patent on this idea by trying to patent the “process” of generating and maintaining the record and calculating the proper numbers.³⁷

When the patent was challenged in court, the Federal Circuit held that the claimed invention stated a process within the scope of patentable subject matter.³⁸ The patent system was broad enough to include merely “electronic” processes, the court concluded, and any method of “inputting numbers, calculating numbers, outputting numbers, and storing numbers” pursuant to an algorithm could suffice.³⁹ Further, the simple calculations transformed input into useful output - a number used for billing - satisfying the utility requirement.⁴⁰ So, according to *Excel*, any computerized algorithm that converts some kind of input into some kind of output can be a patentable process. Again, this is just the all or nothing approach that we would expect from the one-step framework. Faced with the choice of either including or excluding the entire world of

34. 172 F.3d 1352 (Fed. Cir. 1999).

35. *Id.* at 1354.

36. *Id.*

37. *Id.*

38. *Id.* at 1358.

39. *Id.* at 1359 (quoting *In re Alappat*, 33 F.3d 1526, 1544 (Fed. Cir. 1994)).

40. *Id.*

computerized algorithms from the reach of the patent laws, the Federal Circuit has embraced the inclusive approach that welcomes any computerized algorithm.⁴¹

Read together, *State Street* and *Excel* articulate an approach to the one-step framework of classical patent law that is at once perfectly logical and quite startling. Rather than acknowledge the conceptual difficulty created by the general purpose computer's capacity to divide the one-step process into two, the Federal Circuit has simply merged the two steps together when the invention is claimed as a machine, and held that the second step alone is patentable when claimed independently as a process.

VI. CONCLUSION

The interaction between computers and the patent system offers a useful example of how new technologies can change the assumptions upon which preexisting laws are based. As we have seen, the new two-step computer technology doesn't quite fit the old one-step patent law. Absent legislative action, the courts will do their best and try to squeeze the new facts into the old law, resulting either in overly broad protection for computerized algorithms, or very little. And no matter which way they come out, it will be easy to see that what the courts have done is, in fact, a squeeze.

41. The court in *Excel* suggested that this inclusive approach is preferable to the exclusive alternative primarily because it is "responsive to the needs of the modern world." *Id.* at 1356. Presumably this refers primarily to the needs of modern-world patent owners.

