Claiming Electronic and Software Technologies: The Effect of the Federal Circuit Decisions in *Alappat*, *Warmerdam*, and *Lowry* on the Claiming of Mathematical Algorithms and Data Structures

Brian Richard Yoshida
*Kirkpatrick & Lockhart, L.L.P.*

Follow this and additional works at: https://digitalcommons.law.buffalo.edu/buffalolawreview

Part of the [Computer Law Commons](https://digitalcommons.law.buffalo.edu/buffalolawreview)

**Recommended Citation**


This Article is brought to you for free and open access by the Law Journals at Digital Commons @ University at Buffalo School of Law. It has been accepted for inclusion in Buffalo Law Review by an authorized editor of Digital Commons @ University at Buffalo School of Law. For more information, please contact lawscholar@buffalo.edu.
Claiming Electronic and Software Technologies: The Effect of the Federal Circuit Decisions in Alappat, Warmerdam, and Lowry on the Claiming of Mathematical Algorithms and Data Structures

BRIAN RICHARD YOSHIDA†

INTRODUCTION

The rapid emergence of the computer era has motivated developers to seek patent protection for their computer program-related inventions.¹ Since the 1950s, patents for inventions using stored program control have existed, though the disclosures often did not highlight the software aspects of the inventions.² Since the late 1960s and early 1970s, the United States Patent and Trademark Office (hereinafter USPTO) and the courts have struggled to resolve which computer program-related inventions represent patentable subject matter and which fall under the exceptions to patentability relating to laws of nature and ideas.³

† Associate, Kirkpatrick & Lockhart, L.L.P., Pittsburgh, PA; B.S.; 1991, Rochester Institute of Technology (Mechanical Engineering); M.S., 1992, Rochester Institute of Technology (Mechanical Engineering); J.D., 1996, State University of New York at Buffalo School of Law.

1. Computer program-related inventions is a phrase used to describe computer implemented processes involving the execution of a computer program and programmed computers.


3. See generally Application of Prater, 415 F.2d 1378 (C.C.P.A. 1968), reh'g, 415 F.2d 1393 (C.C.P.A. 1969) (where the patent application for a method of processing spectrographic data to produce a quantitative spectrographic analysis of qualitatively-known mixtures by which unknown component concentrations could be determined with minimum error was rejected); Application of Bernhart, 417 F.2d 1395 (C.C.P.A. 1969) (where the patent application for a method of and apparatus for automatically making a twodimensional portrayal of a three-dimensional object from any desired angle and distance and on any desired plane of projection was accepted); Application of Mahony, 421 F.2d 742 (C.C.P.A. 1970) (where the patent application for “synchronizing circuit” for automatically synchronizing receiver of digital information was accepted); and Application of Musgrave, 431 F.2d 882 (C.C.P.A. 1970) (where the patent application relating to “Corrections for Seismic Data Obtained from Expanding-Spread” was accepted).
In the last twenty years, greater attention has focused on the possibility of obtaining patent protection. This is due to the explosion of microprocessor and computer technology and the accompanying increase in the importance of the methodology of programming such computers. The difficulty in resolving the debate between the courts and the USPTO stems from inherent problems conceptualizing the unique nature of computer program-related inventions. Computer program-related inventions are said to "defy the conceptual molds provided by the traditional patent and copyright systems." A computer program-related invention can be described as a process designed to accomplish a certain task, as an apparatus or machine to perform a certain task, or as something in between. At least one commentator sees this difficulty as a reason to deny patent protection for all computer program-related inventions.

However, new and useful computer program-related inventions and algorithms, including mathematical algorithms, should constitute subject matter eligible for patent protection as processes or machines. Yet, prior to the Federal Circuit decisions in In re Alappat, In re Warmerdam, and In re Lowry, the state of the law had been that mathematical algorithms "as such" or "in the abstract" do not constitute patentable subject matter. This was true in theory, but in fact a large number of


7. Since this Article will continually discuss "algorithms," and in particular "mathematical algorithms," the term should be defined. However, the definition of an algorithm has caused much confusion in patent law. See Michael C. Gemignani, Should Algorithms Be Patentable?, 22 JURIMETRICS J. 326, 326 (1982) ("No one can question that the notion of an algorithm has given the U.S. Supreme Court no little trouble"). The definitional problems are due in large part to the Supreme Court's decision in Gottschalk v. Benson, 409 U.S. 63 (1972), and the developments thereafter, and are more appropriately discussed in the algorithm and case-law analysis parts of this article. See infra text accompanying notes 82-89.

8. 33 F.3d 1526 (Fed. Cir. 1994).
9. 33 F.3d 1354 (Fed. Cir. 1994).
10. 32 F.3d 1579 (Fed. Cir. 1994).
11. As will be discussed below, the Supreme Court's 1972 decision in Benson is the
patents were obtained on what were essentially computer programming concepts. While the patents used claim language referring to "apparatus," "systems," "methods," and the like, they covered algorithms implemented on a computer to solve various problems. These pertained to (1) the internal operation of the computer, (2) information processing, and (3) computer interfacing with "physical" processes, such as manufacturing.

This article explains how the Federal Circuit decisions in Alappat, Warmerdam, and Lowry were essentially acknowledgments of the current practice of patent practitioners in claiming the invention as an application of the algorithm directed to a specific device or apparatus, as opposed to the algorithm itself. It argues that the time is ripe for the USPTO and the courts to expand patent protection to algorithms and other computer program-related inventions on a broader scale to keep pace with technological advances. This will assure that deserving inventors are afforded the benefits of patent protection, since it is obvious that Congress will not be providing the direction in this area that the USPTO and the courts had anticipated.

Part I provides an introduction to algorithms by defining and demonstrating their practical applicability, discussing the misleading effects of the Supreme Court's definition of them in Gottschalk v. Benson,12 and assessing the various methods of protecting them as a form of intellectual property. Part II discusses the statutory subject matter requirements of patentability for algorithms and computer program-related inventions and examines the historical background of how the USPTO and the courts interpreted them. Part III examines the Alappat, Warmerdam, and Lowry decisions, and Part IV assesses their impact on the claiming of algorithms and computer program-related inventions, providing specific suggestions on how to

---


Despite [Diamond v. Diehr], programmers have not been beating a path to the Patent Office looking for protection. Why? Aside from the time and money constraints, most programmers want to be able to protect the very thing the Supreme Court has said is not protectable, namely, the algorithm . . . Other than rubber-curing programs [held patentable in Diehr] what software is patentable? Because of the shaky state of the patent law regarding software, there is no definitive answer . . .

Id.

draft claims that will meet the statutory subject matter requirements as interpreted by the USPTO and the courts. The article concludes that the Federal Circuit's recent decisions in Alappat, Warmerdam, and Lowry are a continued expression of the Supreme Court's liberal views toward patentable subject matter and algorithms in Diamond v. Diehr, allowing algorithms to be patentable if they are reduced to some type of practical application.

I. Algorithms

A. Definition of Algorithms

Just what is an algorithm? Prior to the Supreme Court's decision in Benson, there was no patent law definition of an algorithm. Nevertheless, but for the nonpatentability rule of Benson, there would be no need for such a definition. The literature on mathematics and computer science offers a number of definitions of an algorithm that are different in their form, but are mostly consistent in their substance. Given a problem, and a device used to solve the problem, an algorithm is the precise characterization of a method of solving the problem, presented in a language comprehensible to the device. Other definitions, which are simple, broad and informal, include "a recipe or specific set of rules or directions for performing a task," or "a set

13. 450 U.S. 175, 182 (1981) (holding that Congress intended patentable subject matter to include "everything under the sun that is made by man").
   The word did not appear in Webster's New World Dictionary as late as 1957; we find only the older form "algorism" with its ancient meaning, i.e. the process of doing arithmetic using Arabic numerals. In the middle ages, abacists computed on the abacus and algorists computed by algorism. Following the middle ages, the origin of this word was in doubt . . . . Finally, historians of mathematics found the true origin of the word algorism: it comes from the name of a famous Persian textbook author, Abu Ja'far Mohammed ibn Musa al-Khowarizmi (c. 825). . . . Khowarizm is today the small Soviet city of Khiva . . . .
   Id. Gradually the form and meaning of "algorism" became corrupted; as explained by the Oxford English Dictionary, the word was "erroneously refashioned" with the word arithmetic. The Oxford English Dictionary 313 (2d ed. 1959).
15. Unfortunately, the Benson court used the general term "algorithm" when it referred to a specific type of algorithm, one that solved a mathematical problem. As a result, this case raised questions in subsequent appeals of rejections by the USPTO about the meaning of the term algorithm.
17. MICHAEL MACHTEY & PAUL YOUNG, AN INTRODUCTION TO THE GENERAL THEORY OF ALGORITHMS 1 (1978).
of formal directions for obtaining the required solution."\(^{18}\) Another definition which is more formal is as follows:

A method of solution for problem P on device M is a description in a language comprehensible to M of discrete steps performable by M and an ordering of these steps, such that given proper data, if M performs the prescribed steps in the prescribed order, a solution to the problem P will result, if one exists. A method of solution will be called a semi-algorithm for P on M if the solution to P (if one exists) appears after the performance of finitely many steps. A semi-algorithm will be called an algorithm if, in addition, whenever the problem has no solution the method enables the device to determine this after a finite number of steps and halt.\(^{19}\)

All computer program-related inventions are algorithms. An algorithm may be expressed in different ways depending on the device used to carry it out. The steps of an algorithm must be written to be understood and executed by the device used. Where the device is a computer, the algorithm may be expressed in the form of a computer program-related invention. The same algorithm may be written for a human to carry out, but the language used would be different. "While the algorithm, properly formalized, may not be the only way to solve problems, it appears to be essentially the only way that the human intellect in its present stage of development can comprehend."\(^{20}\)

The above definitions all come close to equating an algorithm with a "process" in the patent sense of a sequence of specifically defined operations to accomplish a useful result. Five important features of an algorithm differentiate algorithms from other concepts of problem solving (such as recipes, processes, methods, techniques, procedure, or routines):

1. Finiteness. An algorithm must always terminate after a finite number of steps.
2. Definiteness. Each step of an algorithm must be precisely defined; the actions to be carried out must be rigorously and unambiguously specified for each case.
3. Input. An algorithm has zero or more inputs, i.e., quantities which are given to it initially before the algorithm begins. These inputs are taken from specified sets of objects.
4. Output. An algorithm has one or more outputs, i.e., quantities which have a specified relation to the inputs.

\(^{18}\) MARK A. AISERMAN, ET AL., LOGIC, AUTOMATA, AND ALGORITHM 305 (1971).

\(^{19}\) ROBERT R. KORFHAGE, LOGIC AND ALGORITHMS WITH APPLICATIONS TO THE COMPUTER AND INFORMATION SCIENCES 89 (1966). The author makes it clear that a "device" can include a human being. Id. at 91.

\(^{20}\) KORFHAGE, supra note 16, at 53.
Effectiveness. An algorithm is also generally expected to be effective. This means that all of the operations to be performed in the algorithm must be sufficiently basic that they can in principle be done exactly and in a finite length of time by a man using pencil and paper.  

Many algorithms also appear to have a recursive feature, where one or more of the steps entails going back and repeating one or more of the prior steps.

B. Benson's Misleading Definition of Algorithms

In Benson, the Supreme Court recited a definition of an algorithm:

A procedure for solving a given type of mathematical problem is known as an "algorithm." The procedures set forth in the present claims are of that kind; that is to say, they are a generalized formulation for programs to solve mathematical problems of converting one form of numerical representation to another. From the generic formulation, programs may be developed as specific applications.

The Court erred in two respects: (1) by implying that algorithms relate to or solve only "mathematical problems", and (2) in characterizing the method involved in Benson as directed to "mathematical problems." It is indisputable that algorithms

21. Knuth, supra note 14, at 4-6. By the reference to pencil and paper, this author seems to assume that an algorithm is primarily for information or number processing. Such is not necessarily the case. See infra note 23. Another work lists three "empirical properties" that have been found to be present in all algorithms constructed so far:

(a) Determinacy. The procedure is specified so clearly and precisely that there is no room for arbitrary interpretation. A procedure of this kind can be communicated to another person by a finite number of instructions. The operations described by these instructions do not depend on the whim of the operator and constitute a determinate process which is completely independent of the person carrying it out.

(b) Generality. An algorithm is applicable to more than just one specific problem: it is used for solving a class of problems, with the procedural instructions valid for any particular set of initial data.

(c) Efficacy. This property, sometimes called the directionality of an algorithm, means that application of an algorithmic procedure to any problem of a given kind will lead to a 'stop' instruction in a finite number of steps, at which point one must be able to find the required solution.

Aiserman, supra note 18, at 308-09.

22. For a discussion of the theoretical associations of recursiveness with algorithms, see MacIntey & Young, supra note 17, at 2 ("[A]ll evidence indicates that the class of partial recursive functions is exactly the class of effectively computable functions; that is, that the partial recursive functions are exactly the functions which can be computed by finite procedures, algorithms, or computer programs.")

often are devised in order to solve problems of a mathematical nature. The method at issue in Benson meets all of the above definitions of an algorithm (including the five features). However, is the method one for solving a “mathematical problem” as the Court states? The Court held that the claimed method would wholly preempt a mathematical formula, and would in effect be a patent on an idea, which would make the method unpatentable. Unfortunately, the Court used the general term algorithm to refer to a specific type of algorithm, one that solved a “mathematical problem.” The imprecision of the Court in characterizing the algorithm before it created uncertainty as to the scope of the exclusionary rule that it upholds and led to many questions in subsequent appeals of rejections by the USPTO about the meaning of the term algorithm.

24. Perhaps the most famous is the algorithm derived from Euclid for finding the greatest common divisor of two positive integers $a$ and $b$. One text relates the algorithm as follows:

1. Compare $a$ and $b$ ($a = b$, or $a < b$, or $a > b$). Go on to 2.
2. If $a = b$ then either is the greatest common divisor. Stop the computation. If $a \neq b$ go on to 3.
3. Subtract the smaller from the larger number and write down the subtrahend and the remainder. Go to the next instruction.
4. Assign symbol $a$ to the subtrahend, and symbol $b$ to the remainder. Return to direction 1.

The procedure is repeated until $a = b$. Then the computation is stopped.

Aiserman, supra note 18, at 306.


The first algorithms were invented to solve numerical problems such as multiplying numbers, finding the greatest common divisor, calculating trigonometric functions and so on. Today non-numerical algorithms are of equal importance; they have been devised for tasks such as finding the smallest element in a sequence, searching for a given word in a text, scheduling events and sorting data into some specified order.

Id. See also Korfhage, supra note 16, at 91 (giving an algorithm for “travers[ing] a simple maze without loops”); Goodman, supra note 4, at 153 n.32 (giving an algorithm for a “safe procedure for lifting a cup of coffee off a saucer”).

26. See, e.g., In re Toma, 575 F.2d 872, 875 (C.C.P.A. 1978) (method for “translation between source and target natural languages using a programmable digital computer system” did not involve a “mathematical algorithm” in the Benson sense); In re Freeman, 573 F.2d 1237, 1246-47 (C.C.P.A. 1978) (computer-based control system for typesetting alphanumeric information, including mathematical symbols, is not an algorithm in the Benson sense; even though the applicant himself characterized his invention as a “local positioning algorithm,” the claimed method contained “no mathematical calculations, equations, or formulae”).

27. See discussion infra part III.
C. Protection of Algorithms

Protection for computer program-related inventions, which always consist of algorithms, is available through copyrights, trade secrets, and patents. Each form affords the inventor unique protections, but the specific remedial protections that patents provide has made it the preferential form of protection.

1. Copyrights. Copyrights afford computer program-related inventions only limited protection. A copyright protects only the expression of an idea, not the idea itself. For example, the Supreme Court has held that a copyrighted book did not protect the author’s novel bookkeeping system, but only prevented the unauthorized copying of his expression of these ideas—the book itself. Thus, a copyright would protect computer program-related inventions against unauthorized copying of the program steps but would not prevent another computer programmer from developing a different program that employs the same algorithm to accomplish the same result. Furthermore, because copyrights prevent only the unauthorized copying of the program steps, they fail to afford a remedy against one who independently authors a similar work, or who copies the original work with substantial changes in it. Due to these limitations, copyrights do not adequately protect programs employed in computer program-related inventions. Patents, on the other hand, specifically

29. Id. at 104. Congress codified this concept in 17 U.S.C. § 102(b) (1994), which states that copyright protection does not “extend to any idea, procedure, process, system, method of operation, concept, principle, or discovery, regardless of the form in which it is described, explained, illustrated, or embodied in such work.”
30. See, e.g., Bleistein v. Donaldson Lithographing Co., 188 U.S. 239, 249 (1903) (stating that there is no copyright infringement if one paints an identical portrait using the same model; but one is not free to copy another’s portrait); R. Dakin & Co. v. Charles Offset Co., 441 F. Supp. 434, 439 (S.D.N.Y. 1977) (holding that plaintiff is entitled to recovery for copyright infringement upon a prima facie showing of copying and absent defendant’s showing of independent creation); Fred Fisher, Inc. v. Dillingham, 298 F. 145, 147 (S.D.N.Y. 1924) (stating that one who independently writes or composes the precise combination of copyrighted words or notes is not guilty of copyright infringement). See also 17 U.S.C. § 106 (setting forth the exclusive rights granted to authors). In other words, copyright can be thought of as requiring only “originality” whereas patents must be not only “original”—not copied—but also “novel.” Compare 35 U.S.C. § 102 (1994) with 17 U.S.C. § 102 (1982). Therefore, one must be the first inventor of the subject matter sought to be patented.
31. Copyright infringement requires actual appropriation of language, see Baker v. Selden, 101 U.S. 99, 103-04 (1879), or substantial similarity to the original work, see Gross v. Seligman, 212 F. 930, 931-32 (2d Cir. 1914).
protect an invention's underlying concept.\textsuperscript{32} Although the duration of patent protection is much shorter than that of a copyright, the scope of the subject matter protected is broader.

2. Trade Secrets. A trade secret can consist of "any formula, pattern, device, or compilation of information which is used in one's business, and which gives him an opportunity to obtain an advantage over competitors who do not know or use [the secret]."\textsuperscript{33} Trade secret protection prevents unauthorized disclosure or use of information that has been made available only in confidence.\textsuperscript{34} However, the proprietor of a trade secret has an adequate remedy for unauthorized use of the trade secret only if there has been some misappropriation; for example, if the secret was stolen\textsuperscript{35} or if a confidential relationship or contract was breached.\textsuperscript{36} There is no remedy if the technology is appropriated through lawful means, such as independent discov-

\begin{itemize}
\item [\textsuperscript{33}] Restatement of Torts § 757 comment b (1939). State and federal courts have adopted the Restatement definition of a trade secret even though the second Restatement of Torts no longer includes any trade secret provisions. See, e.g., Aronson v. Quick Point Pencil Co., 440 U.S. 257, 266 (1979) (stating that "the most commonly accepted definition of trade secrets is restricted to confidential information which is not disclosed in the normal process of exploitation" and citing the Restatement's definition of a trade secret); University Computing Co. v. Lykes-Youngstown Corp., 504 F.2d 518, 535 (5th Cir. 1974) (upholding a finding that a computer program was within the Restatement's definition of a trade secret).
\item [\textsuperscript{34}] Restatement of Torts § 757 comment b (1939). State and federal courts have adopted the Restatement definition of a trade secret even though the second Restatement of Torts no longer includes any trade secret provisions. See, e.g., Aronson v. Quick Point Pencil Co., 440 U.S. 257, 266 (1979) (stating that "the most commonly accepted definition of trade secrets is restricted to confidential information which is not disclosed in the normal process of exploitation" and citing the Restatement's definition of a trade secret); University Computing Co. v. Lykes-Youngstown Corp., 504 F.2d 518, 535 (5th Cir. 1974) (upholding a finding that a computer program was within the Restatement's definition of a trade secret).
\item [\textsuperscript{35}] Restatement of Torts § 757 comment b (1939). State and federal courts have adopted the Restatement definition of a trade secret even though the second Restatement of Torts no longer includes any trade secret provisions. See, e.g., Aronson v. Quick Point Pencil Co., 440 U.S. 257, 266 (1979) (stating that "the most commonly accepted definition of trade secrets is restricted to confidential information which is not disclosed in the normal process of exploitation" and citing the Restatement's definition of a trade secret); University Computing Co. v. Lykes-Youngstown Corp., 504 F.2d 518, 535 (5th Cir. 1974) (upholding a finding that a computer program was within the Restatement's definition of a trade secret).
\item [\textsuperscript{36}] Restatement of Torts § 757 comment b (1939). State and federal courts have adopted the Restatement definition of a trade secret even though the second Restatement of Torts no longer includes any trade secret provisions. See, e.g., Aronson v. Quick Point Pencil Co., 440 U.S. 257, 266 (1979) (stating that "the most commonly accepted definition of trade secrets is restricted to confidential information which is not disclosed in the normal process of exploitation" and citing the Restatement's definition of a trade secret); University Computing Co. v. Lykes-Youngstown Corp., 504 F.2d 518, 535 (5th Cir. 1974) (upholding a finding that a computer program was within the Restatement's definition of a trade secret).
\end{itemize}
cery, reverse engineering, or divulgence by the trade secret holder without maintaining confidentiality. In contrast, a patent owner receives an exclusive right to prevent others from making, using, or selling the invention if the patent owner can show that the inventor was the first inventor of the technology patented. The patent holder therefore has an adequate remedy even if a subsequent inventor independently develops the same invention.

3. Patents. Patent protection particularly appeals to developers of computer program-related inventions. A patent provides protection for the concept of an invention unlike a copyright which merely protects the expression of the underlying concept. A patent allows the holder to exclude others from making, using, or selling the invention for seventeen years. The patent holder is also protected against anyone who independently but subsequently develops the same invention. A patent affords the owner of a computer program-related invention with adequate remedies for infringement. A patent owner must sim-

38. Kewanee, 416 U.S. at 476; E.I. duPont, 431 F.2d at 1015. The Supreme Court has described reverse engineering as "starting with the known product and working backward to divine the process which aided in its development or manufacture." Kewanee, 416 U.S. at 476.
40. See 35 U.S.C. § 102 (1994). This section requires that an invention be "novel" in order to be patented and describes ways in which rights to a patent can be lost.
41. See DAVID BENDER, COMPUTER LAW: SOFTWARE PROTECTION 3-1 (1996) (stating that patent protection is useful if a computer-related invention's value resides in its algorithm). Some commentators, however, have expressed doubts about whether the patent system can adequately protect computer programs. See, e.g., Michael C. Gemignani, Legal Protection for Computer Software: The View From '79, 7 Rutgers Computer & Tech. L.J. 269, 301-12 (1980); But see, Arthur J. Keeffe & Terry G. Mahn, Protecting Software: Is It Worth All the Trouble?, 62 A.B.A. J. 906, 907 (1976)(arguing for a "federal trade secret law" which would protect authors of software programs like "patent and copyright laws now do for most other types of intellectual property").
42. See supra notes 26-30 and accompanying text.
45. Patent infringement involves the unauthorized making, using, or selling of a patented invention within the territorial boundaries of the United States during the term of the patent. 35 U.S.C. § 271(a) (1994 & Supp. I 1995). A patent holder may sue an infringer for relief in the appropriate federal court, id. § 141, and ask the court for an injunction to prevent continued infringement and an award of damages. Id. §§ 283-284. The accused infringer may raise the defense of patent invalidity or non-infringement. A court will not find infringement unless the defendant's alleged infringing activities fall within the scope of the patent claim. See, e.g., Kimberly-Clark Corp. v. Johnson & Johnson, 745 F.2d 1437, 1457-58 (Fed. Cir. 1984); ACS Hosp. Sys., Inc. v. Montefiore
ply show that the infringer made, used, or sold something that fell within the scope of a patent claim. Because patent protection provides absolute ownership rights, the owner need not prove unauthorized copying, as is the case with copyrights, nor misappropriation, as is required under trade secret law. The patent laws provide a remedy by civil action and afford damages designed to compensate the patent holder for infringement.

Congress has expressly permitted the licensing of patents, rendering patent protection not only useful, but also profitable. Assuming a computer program-related invention is entitled to patent protection under the statutory requirements of the Patent Act, patents provide protection that is superior to either copyright or trade secret protection.

II. PATENT PROTECTION FOR COMPUTER PROGRAM-RELATED INVENTIONS

A. Statutory Subject Matter—35 U.S.C. § 101

To obtain a patent, an applicant must establish that the subject matter of the invention falls within one of the four statutory categories of invention defined in Section 101 of the Patent Act of 1952: "[w]hoever 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." The categories include a new use of a known process, etc.


47. See supra text accompanying notes 28-31.
48. See supra notes text accompanying notes 33-36.
49. 35 U.S.C. §§ 281, 284 (1994) (providing for "no . . . less than a reasonable royalty for the use made of the invention by the infringer").
50. See id. § 261 (indicating that patents have the attributes of personal property and, therefore, can be freely assigned or conveyed); see also id. § 271(d)(2) (condoning the use of licensing agreements).
51. 35 U.S.C. § 101 (1994). As the Supreme Court has stated, "[N]o patent is available for a discovery, however useful, novel, and nonobvious, unless it falls within one of the express categories of patentable subject matter of 35 U.S.C. § 101." Kewanee Oil Co. v. Bicron Corp., 416 U.S. 470, 483 (1974). The invention in addition must be novel and must meet the nonobviousness requirement set forth in Section 103. According to that section, an invention may not be patented "if the differences between the subject matter sought to be patented and the prior art are such that the subject matter as a whole would have been obvious at the time the invention was made to a person having ordinary skill in the art to which said subject matter pertains." 35 U.S.C. § 103 (1994). The test for the presence of nonobviousness is three-part; the level of ordinary skill and the scope of the prior art must be determined; the differences between the claims at issue
machine, manufacture, composition of matter, or material. If an invention does not fall within one of the categories in Section 101, it is not patentable.

A process is an act or series of acts which produces a desired result. In Cochrane v. Deener, the Supreme Court set forth a classic definition of "process." A machine is a device that has relatively moveable parts and performs a useful operation. Proper statutory claims may include those drawn to the machine itself, to a part of a machine or to a new combination of well-known elements. An article of manufacture is generally defined as any tangible object, other than a machine or composition of matter, that is man-made and not found in substantially the same form in nature. Mere printed matter does not qualify as statutory subject matter under Section 101. Most computer program-related invention claims have been drafted either as a process describing a set of actions to be performed on or by specific combinations of means plus function elements, or as a component of a new machine, in order to satisfy the requirement of Section 101.

The use of the phrase "whoever invents" in Section 101 requires that the claimed invention be man-made, and lays the foundation for the doctrine that phenomena of nature, mental processes, and abstract intellectual concepts are not patentable. Under this theory, discoveries of scientific principles are not patentable. "The rule that the discovery of a law of nature and the prior art must be ascertained; and then the obviousness of those differences to a person of ordinary skill must be decided. Graham v. John Deere Co., 383 U.S. 1, 17 (1966). The ordinary skill is not that of a layman but that of a person reasonably skilled in the art. Dann v. Johnston, 425 U.S. 219, 229 (1976) (quoting Graham v. John Deere Co., 383 U.S. 1, 37 (1966)).

52. 35 U.S.C. § 100(b) (1994).
53. 94 U.S. 780 (1876).
54. Id. at 788 ("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").
56. See, eg., In re Gulack, 703 F.2d 1381 (Fed. Cir. 1983).
57. See 35 U.S.C. § 112 (1994), which provides:
An element in a claim for a combination may be expressed as a means or step for performing a specified function without the recital of structure, material, or acts in support thereof, and such claim shall be construed to cover the corresponding structure, material, or acts described in the specification and equivalents thereof.

Id.
59. For the classic illustration of the distinction between discovery and invention
cannot be patented rests, not on the notion that natural phenomena are not processes, but rather on the more fundamental understanding that they are not the kind of 'discoveries' that the statute was enacted to protect. The courts have applied this rule in excluding a number of subject-matter areas from protection under Section 101, including principles or laws of nature, ideas, mathematical expressions of scientific truths,

see MacKay Radio and Tel. Co. v. Radio Corp. of America, 306 U.S. 86, 94 (1939). For example, Benjamin Franklin's recognition of the electrical nature of lightning was a discovery, whereas his application of the discovery in the form of the lightning rod was an invention.


61. See Funk Bros. Seed Co. v. Kalo Inoculant Co., 333 U.S. 127, 130 (1948). In Funk, the Supreme Court held that the discovery that certain strains of bacteria could be mixed without harmful effect to their properties was merely the handiwork of nature and was therefore unpatentable. Id. The Court stated:

The qualities of these bacteria, like the heat of the sun, electricity, or the qualities of metals, are part of the storehouse of knowledge of all men. They are manifestations of laws of nature, free to all men and reserved exclusively to none. He who discovers a hitherto unknown phenomenon of nature has no claim to a monopoly of it which the law recognizes. If there is to be invention from such a discovery, it must come from the application of the law of nature to a new and useful end.

Id. See O'Reilley v. Morse, 56 U.S. (15 How.) 62, 113 (1853) (stating that a claim directed to the use of electromagnetism without regard to the particular process disclosed in the application is not itself patentable subject matter). Thus, for example:

A new mineral discovered in the earth or a new plant found in the wild is not patentable subject matter. Likewise, Einstein could not patent his celebrated law that E=mc2; nor could Newton have patented the law of gravity. Such discoveries are "manifestations of . . . nature, free to all men and reserved exclusively to none."

Diamond v. Chakrabarty, 447 U.S. 303, 309 (1980) (quoting Funk, 333 U.S. at 130). The application of a law of nature or a mathematical formula to a process or structure, however, may be patentable. Diamond, 450 U.S. at 187-88; The Telephone Cases, 126 U.S. 1, 534 (1888) (finding that Alexander Graham Bell's claims were not an attempt to patent all telephonic uses of electricity).

62. See generally Rubber-Tip Pencil Co. v. Howard, 87 U.S. (20 Wall.) 498, 507 (1874) (stating that an idea itself is not patentable, but a new device in which the idea is put to use may be patented); Le Roy v. Tatham, 65 U.S. (14 How.) 156, 174-75 (1852) (stating that a mere principle, however novel, may not be the subject of a patent; the invention must consist of a practical application and serve some useful purpose); In re Bolongaro, 62 F.2d 1059, 1060 (C.C.P.A. 1933) (holding that a method of producing printed publications from manuscripts is nonpatentable subject matter); Nippon Elec. Glass Co. v. Sheldon, 539 F. Supp. 542, 546 (S.D.N.Y. 1982) (stating that applicant's discovery that x-radiation in excess of .04 millirems per hour given off by television sets is harmful to human beings is an idea that cannot be patented); see also Mitchell P. Novick & Helene Wallenstein, The Algorithm and Computer Software Patentability: A Scientific View of a Legal Problem, 7 Rutgers Computers & Tech. L.J. 313, 316-17 (1980) (explaining that a process consisting only of mental steps is not patentable).

63. See generally Diamond, 450 U.S. at 186 (1981) (stating that an "algorithm, or
and printed matter.\textsuperscript{64}

Thus, if the computer program is viewed as a series of thought processes or as a mathematical algorithm, then it merely consists of mental steps that are not statutory subject matter under Section 101. This was the scenario in the early prosecution of patent claims for computer programs. With the burgeoning of the computer industry in the early 1960s, President Lyndon Johnson set up a committee\textsuperscript{65} to address the patentability of computer programs and to suggest statutory revisions to the Patent Act, which proclaimed that:

The Patent Office now cannot examine applications for programs because of the lack of a classification technique and the requisite search files. Even if these were available, reliable searches would not be feasible or economic because of the tremendous volume of prior art being generated. Without this search, the patenting of programs would be tantamount to mere registration and the presumption of validity would be all but nonexistent.\textsuperscript{66}

\footnotesize{mathematical formula, is like a law of nature, which cannot be the subject of a patent"); Mackay Radio & Tel. Co. v. RCA, 306 U.S. 86, 94 (1938) (stating that “while a scientific truth, or the mathematical expression of it is not a patentable invention, a novel and useful structure created with the aid of knowledge of scientific truth may be” patentable.).

\textsuperscript{64} See generally In re Miller, 418 F.2d 1392, 1396 (C.C.P.A. 1969) (stating that “printed matter by itself is not patentable subject matter”); In re Russell 48 F.2d 668, 669 (C.C.P.A. 1931) (stating that “[t]he mere arrangement of printed matter on a sheet or sheets of paper, in book form or otherwise, does not constitute any new and useful art, machine, manufacture, or composition of matter” (quoting 35 U.S.C. § 31 (1926) (current version at 35 U.S.C. § 101 (1982)))); Boggs v. Robertson, 13 U.S.P.Q. (BNA) 214, 215 (D.D.C. 1931) (stating that printed matter that is independent from the object on which it is arranged is merely an idea reduced to writing and not a manufacture); Ex parte Des Granges, 142 U.S.P.Q. (BNA) 41, 42 (PTO Board of Appeals 1962) (stating that patentability cannot be based solely on printed matter). The copyright laws provide protection for printed matter. See supra notes 26-30 and accompanying text.

Although printed matter itself cannot be patented, certain forms of printed matter can be combined with a physical structure to create patentable subject matter. See, e.g., In re Gulack, 703 F.2d 1381 (Fed. Cir. 1983) (stating that printed matter consisting of printed digits arranged on an underlying band to allow the user to generate prime numbers and perform numerical tricks is functionally related to structural components of the invention and therefore constitutes patentable subject matter); Ex parte Schott, 142 U.S.P.Q. (BNA) 281, 283 (PTO Board of Appeals 1963) (stating that printed matter on a slide rule produced a new functional relationship and therefore constituted patentable subject matter).

\textsuperscript{65} President's Commission on the Patent System, 'To Promote the Progress Of . . . Useful Arts: In an Age Of Exploding Technology (1966) [hereinafter President's Commission Report].

\textsuperscript{66} Id. at 13.
The commission's report\textsuperscript{67} recommended that computer programs be expressly denied patent protection due to the USPTO's lack of adequate facilities and personnel to classify and process such patents.\textsuperscript{68} In turn, the USPTO's 1968 guidelines established that computer programs would not be considered patentable subject matter.\textsuperscript{69}

B. Case Law Developments Concerning the Classification of Computer Program-Related Inventions as Statutory Subject Matter

1. Early Doctrine. The USPTO's early decisions denied patent protection to computer program-related inventions on the basis of the so-called "mental steps doctrine,"\textsuperscript{70} as embraced by the Court of Customs and Patent Appeals (CCPA)\textsuperscript{71} in the case of In re Abrams.\textsuperscript{72} According to that doctrine, any process that consists entirely of mentally executable ideas or operations is not patentable subject matter.\textsuperscript{73}

\textsuperscript{67} Id. at 5.

\textsuperscript{68} Id. at 12. The Commission further addressed the issue of subject matter patentability for computer-related inventions, stating:

Uncertainty now exists as to whether the state permits a valid patent to be granted on programs. Direct attempts to patent programs have been rejected on the ground of nonstatutory subject matter. Indirect attempts to obtain patents and avoid the rejection, by drafting claims as a process, or a machine or components thereof programmed in a given manner, rather than as a program itself, have confused the issue further and should not be permitted.


\textsuperscript{70} See generally Norman D. McClaskey, The Mental Process Doctrine: Its Origin, Legal Basis, and Scope, 55 Iowa L. Rev. 1149 (1970); Katharine P. Ambrose, Comment, The Mental Steps Doctrine, 48 Tenn. L. Rev. 903 (1981). The mental steps doctrine has traditionally been the basis for denying patent protection for claims drawn to pure methods of calculation and mathematical formulae. See, e.g., Lyman v. Ladd, 347 F.2d 482, 483 (D.C. Cir. 1965) (stating that applicant's discovery of a specific relationship between the frequently of a spring suspension element and the gravity induced rate of downward movement of the mechanical elements of the system cannot be the basis for patentability); In re Shao Wen Yuan, 188 F.2d 377, 379-80 (C.C.P.A. 1951) (stating that pure mental steps for computing the profile of an airfoil are outside the scope of patentability).

\textsuperscript{71} The United States Court of Appeals for the Federal Circuit has adopted as precedent the holdings of the CCPA and the Court of Claims and has announced that it will not be bound by the decisions of the other circuit courts. See South Corp. v. United States, 690 F.2d 1368, 1371 (Fed. Cir. 1982) (en banc).

\textsuperscript{72} 188 F.2d 165 (C.C.P.A. 1951).

\textsuperscript{73} See Shao Wen Yuan, 188 F.2d at 379-80.
The mental steps doctrine was an obstacle to the patentability of computer programs. Because a computer program can be thought of as a series of steps for solving a particular problem, it can be thought of as an algorithm. Moreover, if the steps involved mathematical calculations, constituting a mathematical algorithm, it was feared that a patent on the program would be a patent on the algorithm itself, which would wholly preempt the mathematical algorithm, remove it from the public domain, and leave it with no other practical application. This would seem to be contradictory to the exclusionary rule that laws of nature, scientific principles, and mathematical formulas could not be patented.

The USPTO applied the "mental steps doctrine" throughout the 1960s. However, the CCPA began to overturn some of the "mental step" legal barriers to patenting computer-software inventions, replacing them with what Mr. Justice Stevens later referred to disapprovingly as "more expansive principles formulated with computer technology in mind." The CCPA disapproved the "mental steps doctrine" as not being sufficient to answer the complex questions in In re Prater, and concluded that the precedents for the doctrine had been poorly reasoned or misinterpreted. The Prater Court held that although purely mental steps may not be patentable, a process disclosed as being a sequence or combination of steps, capable of performance without mental steps and directed to a category of invention defined in Section 101, may be entitled to a patent even though the process could alternatively be carried out by mental steps. After Prater, the USPTO could no longer use the mental steps doctrine as a basis for denying a patent for a computer program embodying a mathematical algorithm, despite the fact that such a program could be executed through mental calculations.

In In re Bernhart, the CCPA recognized that a computer run by a new and unobvious program differed physically from the computer itself, and suggested that claims drawn to a machine, one of the categories of subject matter listed in Section 101, rather than just the programming steps, could be used to protect computer programs, since the steps were performed by a

---

74. See supra notes 14-19 and accompanying text for the definition of an algorithm.
76. See supra notes 56-62 and accompanying text.
machine and therefore were not mental. This case left little doubt that the "mental steps doctrine" was not applicable to computer program-related inventions except when an allowed patent claim would prohibit all uses of a scientific principle or mathematical equation.

In response to the USPTO's practice of using the "mental steps doctrine," the CCPA developed the "technological arts" test. This held that method or process claims which could be performed on a programmable computer or which were otherwise in the "technological arts" were patentable subject matter because they were not "purely mental." Under this test, the court presumes that applicants for a patent have established a prima facie case that the subject matter of the inventor's application discloses a technical apparatus that implements a process.

2. Supreme Court Intervention. After the CCPA chipped away at the "mental steps doctrine," the Supreme Court took up the challenge of the CCPA's new position in 1972 in the case of Gottschalk v. Benson. The Court focused the debate away from the technological arts and substituted an inquiry into the underlying algorithm utilized in the program and its connection with other physical or mechanical processes that might be the end product of the invention.

In Benson, the CCPA reversed the USPTO's denial of patent protection for Benson's method of converting numerical information from binary-coded decimal numbers into binary numbers, which could then be used to program a computer. The Supreme Court, deciding that the claimed programming algorithm was not a "process" as defined by the Patent Act, reversed the CCPA. The Court reasoned that an algorithm, or mathematical formula, is analogous to a law of nature. It then applied the

---

80. Id. at 1400; see also In re Foster, 438 F.2d 1011, 1016 (C.C.P.A. 1971) (stating that a computer run by a particular program has its memory elements arranged differently than the same computer run by a different program).

81. See Bernhart, 417 F.2d at 1399. The court noted that the apparatus and method claims at issue could only be infringed by a digital computer and therefore could not possibly prohibit all uses of a scientific principle or mathematical equation. Id. at 1399-401.

82. In re Musgrave, 431 F.2d 882 (C.C.P.A. 1970). The test was reaffirmed in In re Foster, 438 F.2d 1011 (C.C.P.A. 1971).


84. 409 U.S. 63 (1972).

85. Id. at 64.

86. Id. at 67.
established rule that a law of nature may not be the subject of a patent and denied patent protection for Benson's invention.87 Noting that the applicant's process claim was “so abstract and sweeping” that it covered all known and unknown uses of the claimed conversion with no specified end use,88 the Court found that the mathematical formula had no other practical application except for its use in a general purpose computer.89 As a result, the Supreme Court refused to affirm the CCPA's determination of patentability. The Court reasoned that a process claim may not be granted patent protection when the process “would wholly preempt the mathematical formula and in practical effect would be a patent on the algorithm itself,”90 and that the discovery of preexisting natural law does not give the discoverer “the right to exclude others from enjoying the benefits derived from the operation of those natural laws.”91 The Court was careful to mention that it did not intend to disqualify all computer programs from patent protection.92 It simply means that a process claim which contains a mathematical formula cannot be drawn so broadly that it covers all uses, known and unknown, of the formula.93 To be patentable, an algorithm must have a specific application to a new and useful end.94

The Supreme Court again responded in Parker v. Flook.95 Flook's program instructed a computer to periodically measure the process variables used in the catalytic conversion of hydrocarbons and employed a previously undiscovered mathematical formula to continually update the allowable limits on those variables, constituting a category of useful, though conventional, applications.

87. Specifically, the Court stated: “He who discovers a hitherto unknown phenomenon of nature has no claim to a monopoly of it which the law recognizes. If there is to be invention from such a discovery, it must come from the application of the law of nature to a new and useful end.” Id. (quoting Funk Bros. Seed Co. v. Kalo Inoculant Co., 333 U.S. 127, 130 (1948)).
88. 409 U.S. at 68.
89. Id. at 71.
90. Id. at 72. The Court defined an algorithm as a “procedure for solving a given type of mathematical problem,” which, like a law of nature, could not be considered patentable subject matter. Id. at 65; see supra note 7.
91. 409 U.S. at 67.
92. See id. at 72-73.
93. Id. at 68.
94. The Court later expressly limited its holding to the facts in Benson in Dann v. Johnson, 425 U.S. 219 (1976), where the application failed on obviousness grounds and the subject matter issue was not reached. In light of Dann, the CCPA subsequently recharacterized patentable software claims as claims for a larger apparatus or process in which software was only part of the invention.
95. 437 U.S. 584 (1978).
post-solution activity.\textsuperscript{96}

The USPTO denied the reasoning in Flook's claims that the only difference from the prior art was the new mathematical formula used for updating. Flook argued that the program did not preempt all use of the algorithm and, under the guide of Benson, should be allowed. The CCPA agreed with Flook and reversed the PTO rejection.\textsuperscript{97}

The Supreme Court, however, denied Flook's claims, broadened its preemption approach taken in Benson, and reaffirmed its definition of algorithm. The Court said that even though Flook did not attempt to preempt all use of his formula, the formula was the only novel portion of Flook's process. Using the figures computed to adjust limits did not make the process patentable.\textsuperscript{98} The Court also stated that if the program output is only used to send the results of a calculation to an operator or screen, tying that activity to a specific end use was merely a recitation of insignificant post-solution activity and thus not patentable. In rejecting Flook's claims, the Supreme Court used what has been called the "point of novelty" approach,\textsuperscript{99} under which claims do not constitute patentable subject matter if the only difference between the applicant's claims and the prior art resided in a mental operation or an otherwise nonstatutory element.

\textit{Flook} made it extremely difficult to patent processes implemented by software. Even if a program did not preempt all use of the formula, as in Benson, if the claim included only post-solution activity, such as the output of data, the claim would not be patentable. The confusion resulting from the dual definition of the term "algorithm" was prevalent in the decisions of the USPTO and the CCPA. The USPTO tended to hold that a computer program expressed in numerical terms was not proper

\textsuperscript{96} Post-solution activity can be described as the steps in a process claim following the solution of the algorithm.
\textsuperscript{97} In re Flook, 559 F.2d 21, 23 (C.C.P.A. 1977), rev'd, 437 U.S. 584 (1978).
\textsuperscript{98} Flook, 437 U.S. 584.
\textsuperscript{99} The Court summarized the "point of novelty" approach as follows: Respondent's process is unpatentable under [Section] 101, not because it contains a mathematical algorithm as one component, but because once that algorithm is assumed to be within the prior art, the application, considered as a whole, contains no patentable invention. Even though a phenomenon of nature or mathematical formula may be well known, an inventive application of the principle may be patented. Conversely, the discovery of such a phenomenon cannot support a patent unless there is some other inventive concept in its application.

\textit{Id.} at 594.
statutory subject matter because the program was inherently an algorithm. The CCPA, however, repeatedly reversed the PTO and ordered such patents to issue.\textsuperscript{100}

3. The Freeman-Walter Test. Following a period of extreme confusion, the CCPA took control and attempted to formulate some concrete criteria for the processing of software claims. The CCPA attempted to reconcile the nature of the algorithm with the claim as a whole by applying the “point of novelty” test, and with the procedure afforded most other scientific processes in a patent application.

The new test became known as the Freeman-Walter two-step test since it was originally set forth in \textit{In re Freeman},\textsuperscript{101} decided before \textit{Flook}, and then modified and put into a form for use in \textit{In re Walter},\textsuperscript{102} a case decided after \textit{Flook}:

First the claim is analyzed to determine whether a mathematical algorithm is directly or indirectly recited. Next, if a mathematical algorithm is found, the claim as a whole is further analyzed to determine if the mathematical algorithm is implemented in a specific manner to define structural relationships between the physical elements of the claim (in apparatus claims) or to refine or limit the claim's steps (in process claims). If it does, the claim is statutory subject matter. If, however, the mathematical algorithm is merely presented and solved by the claimed invention and is not applied in any manner to the physical elements or process steps, no amount of post-solution activity nor limited field of use will render the claim statutory.\textsuperscript{103}

In \textit{Freeman}, the CCPA began to develop its own two-pronged test to determine whether a claim embodying mathematical computations or algorithms constituted proper statutory subject matter under Section 101. The first prong requires the examiner to determine whether the claim at issue directly or indirectly recites an “algorithm” as defined in the \textit{Benson} sense of that term.\textsuperscript{104} If the claim does not recite such an algorithm, it

\textsuperscript{100} See, e.g., \textit{In re Application of Toma}, 575 F.2d 872 (C.C.P.A. 1978); \textit{In re Phillips}, 608 F.2d 879 (C.C.P.A. 1979).

\textsuperscript{101} 573 F.2d 1237 (C.C.P.A. 1978).

\textsuperscript{102} 618 F.2d 758 (C.C.P.A. 1980).

\textsuperscript{103} \textit{In re Pardo}, 684 F.2d 912, 915 (C.C.P.A. 1982).

\textsuperscript{104} 573 F.2d at 1245. The \textit{Benson} Court restricted its consideration to a procedure for solving a given type of mathematical problem (i.e. mathematical algorithms) as opposed to the broader definition of the term algorithm that includes nonmathematical step-by-step procedures for solving a problem or accomplishing some end. See \textit{Benson}, 409 U.S. at 65; \textit{Freeman}, 573 F.2d at 1245. The CCPA reasoned that every process can be characterized as a step-by-step procedure for accomplishing some useful end. Unless Benson concerned itself with only mathematical algorithms, the Court necessarily would
could not "wholly preempt" the algorithm as the Benson Court feared, and as a result, would fall within a statutory category of invention. If the claim does recite an algorithm, the second prong requires the examiner to further scrutinize the claim in order to determine whether the claim, in its entirety, "wholly preempts" the algorithm. Only if the claim satisfies both parts of this analysis would the Benson rule apply, dictating rejection of the claim as being drawn to nonstatutory subject matter. Thus, if a patent applicant could show that his claims did not recite an algorithm or, if one was recited, that the claim would not exclude the public from all uses of that algorithm, then the claim would constitute patentable subject matter.

The second prong of the Freeman test posed a problem of interpretation—when did a claim "wholly preempt" the algorithm? Judge Giles Rich of the Federal Circuit, while serving on the Court of Customs and Patent Appeals, clarified the second step of the analysis in Walter, stating that a claim will fall within Section 101 and constitute patentable subject matter if the mathematical algorithm defines structural relationships between the physical elements in an apparatus claim or refines or limits claim steps in a process claim. If, on the other hand, the claimed invention merely presents and solves the mathematical algorithm, as in Benson and Flook, without applying it to physical elements or process steps, no amount of post-solution activity will render the claim statutory, nor will it be saved by merely reciting the intended use of the mathematical algorithm in the preamble of the claim. Thus, the CCPA harmonized the Freeman-Walter test with the Flook decision, which held that an otherwise nonstatutory claim cannot be transformed into patentable subject matter by merely setting forth some token post-solution activity.

Thus, the preemption approach of Benson was given a narrow interpretation and the end-use test was broadened. The Freeman-Walter test offered a more refined approach to Section 101 statutory subject matter analysis than prior formulations. Pursuant to this approach, the "point of novelty" test is properly

have had to reach the absurd result of reading the word "process" out of Section 101. Id. at 1246.

105. Id. at 1245. The court held that Freeman's claims never reached the second step because they did not recite an "algorithm" in the Benson sense; they failed to set forth "a procedure for solving a given type of mathematical problem." Id. at 1245 (quoting Benson, 409 U.S. at 65 (emphasis added)).

106. 573 F.2d at 1245.

107. 618 F.2d at 767.

108. Id.
considered only after a finding of statutory subject matter is made, not before. As such, although a computer program may be recognized as patentable, it must nevertheless meet the same requirements as other inventions in order to qualify for patent protection.

In *Diamond v. Diehr*, the Supreme Court finally sided with the CCPA and endorsed this approach. This decision also marked the first time the Court recognized a computer program-related invention including a mathematical formula or algorithm within the claimed method as being patentable subject matter. Diehr applied for a patent on a process for molding rubber which ensured that the rubber would always be perfectly cured by utilizing a computer operated program to take constant measurements of the temperature of the mold and apply a known formula to constantly recalculate the remaining cure time. When the computer calculated the remaining cure time to be zero, it automatically opened the mold press. Diehr did not attempt to claim patent protection for the formula itself; he claimed the improved process for curing rubber. In a five to four decision, the Court endorsed the claimed innovation as an improved industrial process, even though it required the use of a digital computer. Since Diehr only attempted to patent the total process and not the formula, algorithm or method of calculation, the Court held that the presence of the formula and the corresponding computer solution did not destroy the statutory subject matter of the process as a whole. Justice Rehnquist, writing for the majority, stated that even if a patent claim for a process contains a mathematical formula, it will satisfy the subject matter requirements of Section 101 if the process, when considered as a whole, implements or applies the formula to perform a function which the patent laws were designed to protect.

The Court also reaffirmed its view regarding the meaning of the word algorithm and again rejected the computer software-algorithm approach. This is made evident by a comparison of *Flook* and *Diehr*. The claims allowed in *Diehr* are quite similar to the ones disallowed in *Flook*. However, the reason the *Diehr* claims were allowed is not because the Court accepted the computer science definition of algorithm over its previously accepted

110. Id. at 181.
111. Id. at 191.
112. Id. at 185-93.
113. Id. at 192.
classical mathematical definition, but rather because in *Flook*,
the only new element was the use of a program to calculate tim-
ing that was previously calculated by hand. In *Diehr*, there had
been no such previous capability in the prior art of measuring
temperature with precision. Thus, the *Diehr* analysis seems to
be more in line with the statutory requirements of Sections 101,
102 and 103, and the treatment given other inventions under
these sections.

*Diehr* did not explicitly open the door for the patenting of
computer programs. Instead, the case established that in in-
stances when a patentable process or apparatus contains a com-
puter program as one of its elements, the process or apparatus
is not rendered unpatentable by the presence of that program. It
also established that patent claims drawn to a computer pro-
gram-related invention must be considered “as a whole”[114] to de-
termine whether they constitute patentable subject matter
under Section 101. However, cases decided subsequent to *Diehr*
show that the substance of a program can indeed be patented,
as long as the claim description includes some reference to a
means for executing the program’s functions.[115]

---

114. See *id.* at 192.

115. In view of the complexity and confusion of the case law in this area, the
USPTO established the following guidelines, extracted from the *Diehr* decision, to in-
struct patent examiners in determining whether a pending claim violates Section 101:

1. The claims must be considered as a whole. It is inappropriate to dissect the
   claims into old and new elements and then to ignore the presence of the old el-
   ements in the analysis . . . . The ‘novelty’ of any element or steps in a process,
or even of the process itself, is of no relevance in determining whether the sub-
ject matter of a claim falls within the 101 categories of possible patentable sub-
ject matter.

2. When a claim containing a mathematical formula implements or applies
   that formula in a structure or process which, when considered as a whole, is
   performing a function which the patent laws were designed to protect (e.g.,
   transforming or reducing an article to a different state or thing), then the
   claim satisfies the requirements of Section 101.

3. When a claim recites a mathematical formula (or scientific principle or phe-
   nomenon of nature), an inquiry must be made into whether the claim is seek-
ing protection for that formula in the abstract. (If the claim does seek
   protection for such a mathematical formula, it would be non-statutory under

4. A mathematical formula as such is not accorded the protection of our pat-
   ent laws . . . . and this principle cannot be circumvented by attempting to limit
   the use of the formula to a particular technological environment . . . . Similarly,
   insignificant post solution activity will not transform an unpatentable principle
   into a patentable process.

5. When a claim as in [*Flook*] is drawn to a method for computing an ‘alarm
   limit’ (which is simply a number, the claim is non-statutory under 35 U.S.C.
§ 101 because *Flook* “sought to protect a formula for computing this number.
Since Diehr, the path to consistent workable standards continued to become more defined. An analysis of subsequent cases shows a greater sense of stability in the application of the standards enunciated in the decision and gives a clearer understanding of the types of software-related claims that will be considered to embrace proper statutory subject matter.\(^1\)

A dramatic example of the line between statutory and non-statutory subject matter comes from the case of In re Abele,\(^1\) where the distinction between software implemented processes and mere calculation was evident in the construction of the claims. The claims in Abele were directed to an improvement in computerized axial tomography (CAT) scanning. The claims recited a method for displaying data from the CAT scan. Some of the claims in the application were found ineligible for patent protection because they were no more than the calculation of a number and a display of the result, albeit in a particular format. Other claims that more specifically recited the device used in conjunction with the algorithm were found to contain patentable subject matter. The CCPA further refined the Freeman-Walter two-step test by holding that if the claims would be otherwise statutory, although inoperative or less useful without the algorithm, the claims present statutory subject matter when the algorithm is included. After the Abele decision, the case became associated with the Freeman-Walter two-step test, which is now known as the Freeman-Walter-Abele test.

4. The Federal Circuit. The Court of Appeals for the Federal Circuit, while not directly considering the patentability of software, explicitly reaffirmed the authority of decisions of its

---

\(^{(6)}\) It is now commonplace that an application of a law of nature or mathematical formula to a known structure or process may well be deserving of patent protection. Diehr, 450 U.S. at 187 (citing Funk Bros. Seed Co. v. Kalo Co., 333 U.S. 127 (1948)); Eibel Process Co. v. Minnesota and Ontario Paper Co., 261 U.S. 45 (1923); Cochrane v. Deener, 94 U.S. 780 (1876); O'Reilly v. Morse, 15 How. 62 (1853); and LeRoy v. Tatham, 14 How. 156 (1852).

\(^{116}\) See, e.g., In re Taner, 681 F.2d 787 (C.C.P.A. 1982) (holding that the process claim and its recited algorithm operated as physical signal processing elements tied to and affecting material objects (analog data) in a specific environment); In re Pardo, 684 F. 2d 912 (C.C.P.A. 1982) (where the claim was essentially a software claim, seemingly unattached to an end processor, but was held not as reciting an algorithm in the mathematical sense of the word as opposed to the computer sense).

\(^{117}\) 684 F.2d 902 (C.C.P.A. 1982).
predecessor, the CCPA,118 and generally followed the reasoning in the Diehr decision.

In In re Grams,119 the Federal Circuit ruled on a claim directed to a method of diagnosing an individual to determine whether the condition of the individual was normal or abnormal. If the condition was abnormal, the cause of the abnormality was determined.120 The first step of the claimed method was to gather the data that were to be used as parameters in the diagnosis. The remaining steps were a mathematical algorithm making use of the collected data. Removing the mathematical language, the court found the only limitations in the claim were the data-gathering steps which merely determined values for the variables used in the mathematical formulae used in making the calculations. The court determined that the data-gathering step was insufficient to render the claim statutory.121

On the other hand, the court held that the mere presence of an algorithm would not automatically preclude the patentability of a claim. If the claim included physical steps (in addition to the algorithm), then the claim may be eligible for patent protection if the physical steps in the claim (without the algorithm) constitute a statutory process and if the algorithm operates on one of the claimed physical steps.122 The court found that the only physical step in Grams' claims was the step of gathering data for the algorithm, and as such, it did not operate to change any aspect of the physical process of gathering data.

In In re Iwahashi,123 the court ruled on the patentability of a computer program-related invention which determined the auto correlation coefficients for use in pattern recognition without the need for expensive multipliers and other complicated circuitry.124 Instead of multiplication, addition and the use of squares were used to determine the coefficients. The sole claim in the application specifically recited an integrated circuit chip used to permanently store information (ROM), in which the values of the squares of certain variables were stored. The other elements in the claim were recited in the means-plus function format.125

119. 888 F.2d 835 (Fed. Cir. 1989).
120. Id. at 836.
121. Id. at 840.
122. Id. at 838.
123. 888 F.2d 1370 (Fed. Cir. 1989).
124. Id. at 1371.
125. Id. at 1372.
The Federal Circuit reversed the holding of the USPTO that the claim was not directed to patentable subject matter. The court relied in part on the fact that the claim included the ROM, a specific piece of apparatus. Therefore, the court found that the claim was directed to a combination of interrelated means within the meaning of Section 101. Relying on Abele, the court found that the fact that the apparatus operated according to an algorithm did not make the claimed invention non-statutory. The court rejected the USPTO’s arguments that the method claim was so broad as to encompass every means for performing the recited functions by referring to 35 U.S.C. § 112, ¶ 6, which states that each means-plus-function definition “shall be construed to cover the corresponding structure, material or acts described in the specification and equivalents thereof.” Based upon the readily recognizable hardware circuit elements from the specification and the provisions of 35 U.S.C. § 122, the court held the claimed invention to be statutory.

III. RETURN TO THE PRIMARY AUTHORITIES

The decisions of In re Alappat, In re Warmerdam, and In re Lowry raised as many new questions as they answered. They left no doubt that the USPTO must apply paragraph 6 to Section 101 determinations, but they offered little concrete guidance as to how that paragraph should be used. While paragraph 6 seems to function well in the context of Sections 102 and 103 to determine the scope of a claim, the dissent in Alappat hinted that the use of paragraph 6 to determine a claim’s scope produces conflicting indications in Section 101 determinations. All three of these decisions appear to ignore the traditional Freeman-Walter-Abele test, and imply that the trilogy of Supreme Court decisions, Gottschalk v. Benson, Parker v. Flook, and Diamond v. Diehr overruled this test. In any event, the emphasis seems to have shifted from determining what a mathematical algorithm is, to whether the claimed invention fits into one of the categories defined by 35 U.S.C. § 101,

126. Id. at 1375.
127. Id.
128. Id.
129. 33 F.3d 1526 (Fed. Cir. 1994).
130. 33 F.3d 1354 (Fed. Cir. 1994).
131. 32 F.3d 1579 (Fed. Cir. 1994).
133. 437 U.S. 584 (1978).
namely a "process," a "machine," a "manufacture," or a "composition of matter."

A. In re Alappat

Just when it seemed that the courts, practitioners, and the USPTO settled into a routine application of the Freeman-Walter-Abele test, a majority of the Federal Circuit decided that the claim at issue in Alappat was statutory subject matter without reference to the test. In Alappat, claim 15 was to "a rasterizer for converting vector list data representing sample magnitudes of an input waveform into anti-aliased pixel illumination intensity data to be displayed on a display means." The claim was in "means plus function" format and so was similar to the claims in In re Noll and In re Freeman.

The majority rested its decision on two rationales. First, the Alappat application clearly describes well known digital circuits which may be correlated to each "means plus function" recitation of the claims in question. Second, an analysis of the language of apparatus claim 15, the sole independent claim, finds that it transforms data.

While differing on the outcome, both the majority and dissent agree on two significant points. First, 35 U.S.C. § 112, ¶ 6 requires the USPTO, even at the examination stage, to interpret "means plus function" and step recitations in view of the corresponding structure and acts described in the specification and equivalents thereof. It appears to be settled that the USPTO must consider the specification in the examination of patent applications as required by paragraph 6 in Section 101, as well as in making Sections 102 and 103 determinations. Second, the majority and dissent based their reasoning on the primal authorities governing statutory subject matter decisions, namely the language of Section 101 and the trilogy of Supreme Court decisions, Benson, Flook, and Diehr. As one would expect, the majority and dissent treat these primary authorities quite differently.

The majority construed Alappat's claim 15 as follows:

135. 33 F.3d at 1541.
136. 545 F.2d 141 (C.C.P.A. 1976).
137. 573 F.2d 1237 (C.C.P.A. 1978).
138. 33 F.3d at 1540.
139. See, In re Donaldson Co., Inc. 16 F.3d 1189 (Fed. Cir. 1994).
A rasterizer [a "machine"] for converting vector list data representing sample magnitudes of an input wave form into anti-aliased pixel illumination intensity data to be displayed on a display means comprising:

(a) [an arithmetic logic circuit configured to perform an absolute value function, or an equivalent thereof] for determining the vertical distance between the endpoints of each of the vectors in the data list;
(b) [an arithmetic logic circuit configured to perform an absolute value function, or an equivalent thereof] for determining the elevation of a row of pixels that is spanned by the vector;
(c) [a pair of barrel shifters, or equivalent thereof] for normalizing the vertical distance an elevation; and
(d) [a read only memory (ROM) containing illumination intensity data, or an equivalent thereof] for outputting illumination intensity data as a predetermined function of the normalized vertical distance and elevation.

The bracketed matter represents the structure disclosed in Alappat's specification which corresponds to these "means plus function" recitations. Paragraph 6 requires claim 15 to be interpreted to cover such structure and equivalents thereof. The majority reasoned that each means claim could be represented in the claim by the structure disclosed, which in this case was two arithmetic logic circuits, a pair of barrel shifters and a ROM. Thus the court concluded that this was a true apparatus claim which defined a "machine" under Section 101.

The dissent, although agreeing with the majority that Alappat's claim 15 covers the structure disclosed, being the "two ALU's, the two barrel shifters, and the ROM" and equivalents thereof, adopted a different argument: "The presence of structure on the face of the claims does not ipso facto make the claimed invention or discovery one of statutory subject matter." To the extent that precedent of the CCPA held that only claims which "wholly preempt" or recite "in its entirety mathematics," i.e., claims which did not recite any structure, were beyond the pale of Section 101, the dissent would hold that such precedent conflicts with Flook. According to the dissent, the Supreme Court in Flook "expressly reversed the court's wholesale preemption test" and "concluded that the claimed discovery was nonstatutory even though the applicant's claim did

140. 33 F.3d at 1541.
141. Id.
142. Id. at 1542.
143. Id. at 1561 (Archer, J., dissenting).
144. Id.
145. Id.
146. Id. at 1562.
not wholly preempt the mathematical function involved."\textsuperscript{147}

With apparent reference to the USPTO's "otherwise statutory" test, the majority asserts that \textit{Diehr} requires that a Section 101 analysis be of the "claim as a whole."\textsuperscript{148} Although the examination of the whole claim is dispositive as to the Section 101 issue, the majority notes that "an analysis wherein one attempts to identify whether any part of a claim recites mathematical subject matter which would not by itself be patentable is not an improper analysis. Such a dissection of a claim may be helpful under some circumstances to more fully understand the claimed subject matter."\textsuperscript{149} For example, an "otherwise statutory" analysis showing that a claim with the mathematical algorithm recitations removed was directed to patentable subject matter, would be convincing particularly to the USPTO. However, as \textit{Alappat} demonstrates, failure of a claim to pass the "otherwise statutory" test is not dispositive.

The majority and the dissent disagreed on how paragraph 6 is to be applied to Section 101 determinations. The majority agreed with \textit{Alappat} that his claim 15 should cover, by virtue of paragraph 6, a programmed general purpose digital computer, i.e., such a computer is an equivalent of the means recited in claim 15. As such, the majority uses paragraph 6 to conclude that claim 15 covers well known digital circuits and is therefore statutory and, at the same time, covers a programmed general purpose computer, even though such computer could not be claimed directly.\textsuperscript{150}

The dissent attacks this reasoning by arguing that "[i]f a programmed general purpose digital computer is not statutory subject matter, then a claim cannot be drawn to that subject matter whether outright or by application of equivalents under 35 U.S.C. [Section] 112, [paragraph] 6."\textsuperscript{151} According to this argument, the finding of equivalency in a programmed general purpose computer proves the nonstatutory nature of \textit{Alappat}'s purported invention. The dissent states that "[p]aragraph 6 of [Section] 112 is not a magical way to expand patent protection into nonstatutory subject matter."\textsuperscript{152}

Underlying the dissent's argument is the recognition of the two-fold nature of paragraph 6. The specific structure described

\textsuperscript{147} Id. at 1561.
\textsuperscript{148} Id. at 1544.
\textsuperscript{149} Id. at 1543, n.21.
\textsuperscript{150} Id. at 1565.
\textsuperscript{151} Id.
\textsuperscript{152} Id.
in the application corresponding to the "means plus function" recitations tends to narrow the claim scope and to render it statutory. Equivalency, on the other hand, tends to broaden the claim and render it nonstatutory. Alappat's argument that his claim 15 would cover by equivalency a programmed general purpose digital computer serves to broaden the claim's scope, but if there is no effective limit to the scope of the "means plus function" recitations, then it would seem that they would be nonstatutory. The practical approach for the applicant, who is going to rely on paragraph 6 for Section 101 purposes, is to spell out in the specification as filed the equivalents or range of equivalents which are intended for the specifically described structure.

Because of the two-fold nature of paragraph 6, the majority's opinion appears to be inconsistent. The opinion first implies that a specification merely describing a programmed general purpose digital computer does not describe "any supporting structure" and then, in a further breath, states that "we have held that such programming creates a new machine, because a general purpose computer in effect becomes a special purpose computer once it is programmed to perform particular functions pursuant to instructions from program software." Both the majority and the dissent accept that Diehr would not permit patent protection for "laws of nature, natural phenomena, and abstract ideas." The majority, however, recognizes a fourth prohibited category, mathematical algorithms, whereas, the dissent would limit its analysis to whether the claimed invention fell into one of the three categories named by Diehr. The majority commented that:

The Supreme Court also has held that certain mathematical subject matter is not, standing alone, entitled to patent protection. (Citations omitted) A close analysis of Diehr, Flook, and Benson reveals that the Supreme Court never intended to create an overly broad, fourth category of subject matter excluded from [Section] 101. Rather, at the core of the Court's analysis in each of these cases lies an attempt by the Court to

---

153. Id. at 1545.
155. 33 F.3d at 1542.
156. Id. at 1582 (Archer, J., dissenting).
157. See, Diehr, 450 U.S. at 191.
explain a rather straightforward concept, namely, that certain types of mathematical subject matter, standing alone, represent nothing more than abstract ideas until reduced to some type of practical application, and thus that subject matter is not, in and of itself, entitled to patent protection.\textsuperscript{158}

The dissent, in refusing to define what constitutes a mathematical algorithm, counters that:

It is unnecessary to discuss what is or is not a "mathematical algorithm," as opposed to being a mathematical "relationship," "formula," "operation," "function," "principle," "theory," or the like. The Supreme Court did not arrive at its holdings in \textit{Benson}, \textit{Flook}, and \textit{Diehr}, discussed \textit{infra}, by creating a new rule about "algorithms" and finding in two cases algorithms and in the other no algorithm. Rather, the holdings are expressly based upon the axioms that abstract ideas, principles, and laws of nature are not patentable subject matter, but that their useful applications may be. Mathematic operations, like ideas and laws of nature, are not useful applications and therefore not statutory subject matter. The hypertechnical distinction between calling something a mathematical "algorithm" versus another mathematic noun is without legal distinction.\textsuperscript{159}

The majority used a "wholly preemption" rationale not unlike the second step of the \textit{Freeman-Walter-Abele} test in holding Alappat's claim 15 statutory. They found that the claim was "limited to the use of a particularly claimed combination of elements performing the particularly claimed combination of calculations to transform, i.e., rasterize, digitized waveforms (data) into anti-aliased, pixel illumination data to produce a smooth waveform."\textsuperscript{160} Furthermore, the majority found that "the claim preamble's recitation that the subject matter for which Alappat seeks patent protection is a rasterizer for creating a smooth waveform is not a mere field-of-use label having no significance."\textsuperscript{161} As such, Alappat's claim 15 was found to define a combination of elements constituting a "machine" within the statutory subject matter defined in Section 101. It was not a disembodied mathematical concept which could be characterized as an "abstract idea," but rather a specific machine to produce a useful, concrete, and tangible result.

The dissent squarely rests its conclusion on the trilogy of Supreme Court decisions, stating "[u]nder \textit{Benson}, \textit{Flook}, and \textit{Diehr} the posing and solution of a mathematic function is non-

\begin{flushright}
158. 33 E3d at 1543.
159. \textit{Id.} at 1555, n.16 (Archer, J., dissenting).
160. \textit{Id.} at 1544.
161. \textit{Id.}
\end{flushright}
statutory subject matter.” This is so even if the mathematical algorithm is carried out on a particular general purpose computer or digital circuitry. In reaching the conclusion that Alappat's claim 15 is no more than the prohibited “posing and solution,” the dissent notes that claim 15 recites merely a “rasterizer” in contrast to an “oscillator” which would include a display, e.g., a cathode ray tube. Focusing on the claim's preamble, the dissent comments “[t]he preamble calls the data ‘anti-aliased pixel illumination intensity data.’ Of course, no matter how many adjectives the claim uses to describe data, data are still data – i.e., pure numbers.

As a practical matter, it may have been preferred to redraft such a “data” recitation as “an electrical signal indicative of the desired value of the anti-aliased pixel illumination intensity.” In the end, the dissent concludes that:

Claim 15 as a whole thus claims old circuitry elements in an arrangement defined by a mathematical operation, which only performs the very mathematical operation that defines it. Rather than claiming the mathematics itself, which of course Alappat cannot do, Alappat claims the mathematically defined structure. But as a whole there is no “application” apart from the mathematical operation that is asserted to be the invention or discovery.

In summary, the Alappat court took a much broader view of Section 101 than the mere application of the traditional Freeman-Walter-Abele test. The court instead relied on the philosophical underpinning of that test. Consequently, the test remains in place as a helpful guide. However, where the test fails, the applicant can still argue the general principles to meet the requirements for statutory subject matter. Even so, in the case of In re Schrader, the dissent asserted that the test was met, yet the claims were rejected as non-statutory subject matter on the basis that no physical quantities were being manipulated. As a matter of fact, the court has suggested that a return to the general principals as enunciated in Diehr may be the best course.
B. In re Warmerdam: Are Data Structures Statutory Subject Matter?

Most program related cases are directed to application programs which interface with data that may be stored in particular data structures. A "data structure" is defined as a "physical or logical relationship among data elements, designed to support specified data manipulation functions." The court in In re Bradley found such structures to be patentable. However, in In re Warmerdam, the USPTO argued that a claim to a "data structure" generated by the method of any of claims 1 through 4 "is not one of the categories of subject matter recited in Section 101, to wit, a process, machine, manufacture, composition of matter, or an improvement thereof.

The Federal Circuit agreed with the USPTO in this case in rejecting claims 1-4 as nonstatutory, but not on the argued rationale that it preempted a mathematical algorithm prescribed by the Freeman-Walter-Abele test. In the trend that began in Alappat, the court comments that it has sought within "Supreme Court guidance . . . to find more precise definitions for the things excluded" from [Section] 101. The court notes that not all of its attempts for greater precision had met with "complete success," for example In re Schrader. In Schrader, the court applied the Freeman-Walter-Abele test finding a mathematical algorithm, namely "summing.

According to the court, the trouble with this often-used test "is that there is no clear agreement as to what is a 'mathematical algorithm,' which makes rather dicey the determination of whether the claim as a whole is no more than that." An alternative to creating these arbitrary definitional terms which deviate from those used in the statute may lie simply in returning to the language of the statute and the Supreme Court's basic principles as enunciated in Diehr, and eschewing efforts to de-

170. 600 F.2d 807 (CCPA 1979).
171. 33 F.3d 1354 (Fed. Cir. 1994).
172. Id. at 1358.
173. Id. at 1358-59. See supra note 129 and accompanying text discussing the significance of the trilogy of Supreme Court decisions.
174. 22 F.3d 290 (Fed. Cir. 1994) (note that Judge Plager not only authored Warmerdam but also Schrader).
175. 33 F.3d at 1359.
176. 22 F.3d at 292.
177. 33 F.3d at 1359 (citing Schrader, 22 F.3d at 292 n.5 and dissent thereto).
scribe nonstatutory subject matter in other terms.\textsuperscript{178}

According to the court, \textit{Warmerdam was a good example of the difficulty in identifying whether a claim, here claim 1, recites an algorithm or not. The USPTO argued that the specification describes the use of a known mathematical procedure, the Hilditch Skeletonization method, acknowledged by the court as being the only practical embodiment.\textsuperscript{179} On the other hand, Warmerdam argues that claim 1 is broad enough to cover the physical measuring of the contour of the object, and does not require the known method described in the specification. The court agreed with Warmerdam and concluded that, "The fact that the claim covers methods which are essentially mathematical in nature, as discussed \textit{infra}, is not dispositive. Claims should be evaluated by their limitations, not by what they incidentally cover."\textsuperscript{180}"

The court felt that there was no need to find the "proper meaning of the label mathematical algorithm,"\textsuperscript{181} but rather states that "the dispositive issue for assessing compliance with Section 101 in this case is whether the claim is for a process that goes beyond simply manipulating 'abstract ideas' or 'natural phenomena.'"\textsuperscript{182} In other words, the proper test is not finding the elusive mathematical algorithm, but rather determining whether the claimed invention, considered as a whole, is one of those three categories prescribed by \textit{Diehr}, namely "laws of nature, natural phenomena, and abstract ideas."\textsuperscript{183} The court concluded that "[t]hese steps describe nothing more than the manipulation of basic mathematical constructs, the paradigmatic 'abstract idea.'"\textsuperscript{184} Since abstract ideas are not statutory subject matter, a data structure defined by those ideas is also nonstatutory subject matter.

The court stated that there is nothing magical about the term "data structure." The court rejected Warmerdam's reliance on \textit{Bradley},\textsuperscript{185} stating that, "[t]he 'data structure' at issue in that case was a physical, interconnected arrangement of hardware and thus embraced by the term 'machine'."\textsuperscript{186} "The same cannot be said of the data structure that is at issue in this case. Con-

\textsuperscript{178.} \textit{Id.}
\textsuperscript{179.} \textit{Id.} at 1360.
\textsuperscript{180.} \textit{Id.} at 1359.
\textsuperscript{181.} \textit{Id.} at 1360.
\textsuperscript{182.} \textit{Id.}
\textsuperscript{183.} \textit{Id.} at 1358.
\textsuperscript{184.} \textit{Id.} at 1360.
\textsuperscript{185.} 600 F.2d 807.
\textsuperscript{186.} 33 F.3d at 1362 (citing \textit{Bradley}, 600 F.2d at 812-13).
trary to Warmerdam's assertion, the phrase does not imply a physical arrangement of the contents of a memory."\textsuperscript{187}

C. In re Lowry: A Data Structure Stored On a Memory is Deemed Patentable

In the recent case of In re Lowry,\textsuperscript{188} the Federal Circuit held for the first time that a claim reciting essentially a memory with data stored thereon, is patentable subject matter. The examiner had rejected Lowry's claims 1-5 as being nonstatutory under Section 101 and as being obvious under Section 103. The USPTO Board agreed that the claims to a particular claimed data structure were within the bounds of Section 101 as an article of manufacture, but rejected them under Section 103. Basically the USPTO took the position that the data structure was printed matter and relied on the holding in In re Gulack,\textsuperscript{189} that "[w]here the printed matter [data structure] is not functionally related to the substrate [memory], the printed matter will not distinguish the invention from the prior art in terms of patentability."\textsuperscript{190} Thus the Board concluded that no patentable weight would be given to the data structure specified in the claims and the claims were thus not patentable over prior data structures with different arrangements.

The Federal Circuit reversed the Board's decision holding that it was an improper extension of the court's printed matter holdings under Section 102 to a rejection under Section 103. Further, the court explains why the printing matter cases are not relevant to the claimed data structures, stating:

The printed matter cases "dealt with claims defining as the invention certain novel arrangements of printed lines or characters, useful and intelligible only to the human mind . . . . The printed matter cases have no factual relevance where "the invention as defined by the claims requires that the information be processed not by the mind but by a machine, the computer . . . . Lowry's data structures . . . are processed by a machine . . . . The printed matter cases have no factual relevance here."\textsuperscript{191}

Even though Lowry deals with an obviousness rejection, it is relevant to this discussion of statutory subject matter. First, it deals with "printed matter" which has, at least previously, been

\textsuperscript{187} Id.
\textsuperscript{188} 32 F.3d 1579 (Fed. Cir. 1994).
\textsuperscript{189} 703 F.2d 1381 (Fed. Cir. 1983).
\textsuperscript{190} Id. at 1385.
\textsuperscript{191} 32 F.3d at 1583 (citing In re Bernhart, 417 1395, 1399 (C.C.P.A. 1969).
considered a category of nonstatutory matter. Second, this case may be important for what it did not hold, namely that the Federal Circuit did not expressly abandon "printed matter" as a category of nonstatutory subject matter, even though both *Alappat* and *Warmerdam* seem to have deliberately avoided definitions of statutory subject matter other than those named in *Diehr* and Section 101 itself. Third, in accordance with this trend, the court demonstrates its preference for the "article of manufacture" definition over the "printed matter" definition of statutory subject matter. Fourth, even though the court decides the fate of claims 1-5 based upon Section 103, it offers comments which seem to be relevant to the Section 101 issue.

Though no statutory subject matter rejection is at issue, the court explains why Lowry's claimed memory is not merely an "abstract idea."

Contrary to the PTO's assertion, Lowry does not claim merely the information content of a memory. Lowry's data structures, while including data resident in a database, depend only functionally on information content. While the information content affects the exact sequence of bits stored in accordance with Lowry's data structures, the claims require specific electronic structural elements which impart a physical organization on the information stored in memory. Lowry's invention manages information. As Lowry notes, the data structures provide increased computing efficiency. Indeed, Lowry does not seek to patent the Attributive data model in the abstract. Nor does he seek to patent the content of information resident in a database. Rather, Lowry's data structures impose a physical organization on the data.192

Next the court demonstrates that the claimed invention of Lowry is really that article of manufacture listed in Section 101:

In Lowry's invention, the stored data adopt no physical "structure" per se. Rather, the stored data exist as a collection of bits having information about relationships between the ADOs [attributable data objects]. Yet this is the essence of electronic structure. In *Bernhart*, this court's predecessor noted:

There is one further rationale used by both the board and the examiner, namely, that the provision of new signals to be stored by the computer does not make it a new machine, i.e., it is structurally the same, no matter how new, useful and unobvious the result. . . . To this question we say that if a machine is programmed in a certain new and unobvious way, it is physically different from the machine without that program; its memory elements are differently arranged. The fact that these physical changes are invisible to the eye should

192. Id.
More than mere abstraction, the data structures are specific electrical or magnetic structural elements in a memory. According to Lowry, the data structures provide tangible benefits: data stored in accordance with the claimed data structures are more easily accessed, stored, and erased. Lowry further notes that, unlike prior art data structures, Lowry’s data structures simultaneously represent complex data accurately and enable powerful nested operations. In short, Lowry’s data structures are physical entities that provide increased efficiency in computer operation.

Though the above comments are on their face directed to the “printed subject matter” issue, they also provide guidance for drafting claims to memories with data structures or perhaps an application program stored therein, so as to avoid Section 101 problems. A claim reciting a memory for storing “mere data” would not be allowed in view of the above comments at least under a “printed matter” rejection, if not a Section 101 rejection. Though something more than “mere data” is required, the operative question is: what is that something? First, that something need not be that “relation between” the data and the memory which was recognized by the Board. Rather, the Federal Circuit only required a “physical organization on the data stored in memory.” Second, to avoid the “printed matter” rejection, the claim must “require” that the stored data be processed by a machine, e.g., a computer.

How does the data structure in Warmerdam relate to the memory/data claim structure approved in Lowry? Both data structures are of a hierarchical configuration. Warmerdam’s data structure comprised a hierarchy of spheres, which Warmerdam admitted to be old; Warmerdam only asserted that his method of generating the spheres was novel. Even if novel, the stored data to define such a hierarchy of spheres may only comprise for each level of the hierarchy a set of coordinates defining the sphere centers and the corresponding radii. Unless the data was stored in some functional relation either to the computer using that data or to the data itself, such sets of coordinates and radii might well be deemed unpatentable “mere

193. Bernhart, 417 F.2d at 1400.
194. Id. at 1583-84.
195. Id. at 1581.
196. Id. at 1583.
197. See Lowry, 32 F.3d at 1583; Bernhart, 417 F.2d at 1399-1400.
data." The dicta in Warmerdam that his machine claim 5 was statutory seems to be distinguishable from the holding in Lowry that his data structure/memory claims 1-5 were patentable. First, Warmerdam deals with Section 101 and Lowry with Sections 102 and 103. Second and more to the point of the foregoing discussion, claim 5 of Warmerdam recites a machine comprising a memory, whereas claims 1-4 of Lowry merely recite a memory for storing his data structure. It is the absence of the machine or computer that renders claims in the Lowry format of significant commercial interest.

The question now becomes, are claims merely to a memory for storing a computer program patentable? Lowry may suggest that if the instructions of the program are arranged in a "physical organization" either between the instructions per se and/or between the instructions and the data to be processed, then a claim to such a structure may pass muster over a "printed matter" rejection under Sections 102 and 103, and presumably over a nonstatutory rejection under Section 101.198

IV. CLAIMING ELECTRONIC AND SOFTWARE TECHNOLOGIES IN LIGHT OF ALAPPAT, WARMERDAM, AND LOWRY

A. The Current Practice

Courts and the Patent and Trademark Office are beginning to show some uniformity in the treatment of software claims, are becoming adept at construing them, and are appearing to be generally at ease under the guide of Diehr. Although adherence to the traditional definition of algorithm has still been subject to some debate, it seems likely that the Diehr analysis will continue to be controlling. For purposes of Section 101 subject matter requirements, the algorithm problem can be avoided by applying some general rules. If the claim embraces a purely mathematical algorithm and is drawn to that algorithm, it is clearly unpatentable subject matter. When a software related claim contains no algorithm or has specific elements that are end use-related, it is proper statutory subject matter under Section 101 and will be subject to the other criteria of patentability.199 Claim drafters should realize that claims must go beyond

198. The USPTO has a policy of rejecting claims which merely recite a memory and a set of instructions stored therein recited in means plus function format. However, some applications have slipped through and have issued. The question now is: what policy will the USPTO adopt with respect to Lowry claims?

199. Among the software-related patents issued in the United States are:

(1) No. 4,823,108, Display System And Memory Architecture And Method
mere calculation, manipulation or insignificant post-solution activity in order to be patentable subject matter. Significantly, mere use in a larger process is not an automatic qualification for subject matter patentability; some end-use application may also be necessary.

Claims that embrace software standing alone as a process can be proper subject matter, but it is wise to tie the process to some means for carrying out the functions performed by the program. Additionally, if the software program itself can be described as an electronic hard-wired configuration of the computer, it may qualify for protection as an apparatus claim drawn solely to the electronic configuration. Important in this regard is the fact that, should the hard-wired version be afforded protection, a competitor's software version could easily be found to be an infringement under the doctrine of equivalents.

Despite qualifying as proper subject matter and passing the test of novelty, usefulness and nonobviousness, applications are denied if the applicant fails to describe the invention "in such full, clear, concise, and exact terms as to enable any person skilled in the art to which it pertains, or with which it is most nearly connected, to make and use the same." Section 112 also requires the applicant to disclose the "best mode" of practicing the invention. These requirements have an impact on the possibility of maintaining trade secret protection that must be care-
fully considered in an overall assessment of how best to protect software.

B. Claim Limitations to Prevent Algorithm Preemption

Where there is a mathematical algorithm invention, it is not patentable as such. Thus, it must be limited so as not to preempt all uses of the algorithm. The question is what must be added to make the algorithm statutory subject matter.

1. Limiting The Field of Use. One possibility is to limit the field of use of the algorithm. However, in *In re Waldbaum*,\(^{203}\) the Court found claims to an algorithm for counting the ones in a data word to be non-statutory, even though the ones represented busy telephone lines. In the Walter case the court held:

The preamble in the claim involved in *Flook*, while limiting the application of the claimed method to “a process comprising the catalytic chemical conversion of hydrocarbons,” did not serve to render the method statutory because the claim, as a whole, was still directed to the solution of a mathematical problem.\(^{204}\)

Illustrative, is the Federal Circuit’s decision in *Arrhythmia Research Technology, Inc., v. Corazonix Corp.*,\(^{205}\) from which it is clear that the “what is being done in the real world” question can be answered in such a way that the claim as a whole is in the statutory subject matter realm. The preamble to claim 1 of the Arrhythmia patent recites: “A method for analyzing electrocardiograph signals to determine the presence or absence of a predetermined level of high frequency energy in the late QRS signal.”\(^{206}\) The elements of the claim further recite: “converting . . . signals . . . applying . . . time segments [signals] . . . to a high pass filter means; [and] determining [a] . . . value of the amplitude of the [signal] output of said filter; and comparing said value with said predetermined level.”\(^{207}\)

At each step of the *Arrhythmia* claim 1, “signals” are explicitly or implicitly recited. Furthermore, the signals are clearly physical signals related to the real world function of electrocardiograph analysis.

---

\(^{203}\) 559 F.2d 611 (C.C.P.A. 1977).

\(^{204}\) *In re Walter*, 618 F.2d 758, 767 (C.C.P.A. 1980).

\(^{205}\) 958 F.2d 1053 (Fed. Cir. 1992).

\(^{206}\) *Id.* at 1055.

\(^{207}\) *Id.* at 1055.
The *Arrhythmia* decision evinces a useful drafting tip: emphasize the real world application being performed and the physical nature of the operations (or "means") being performed (or used); reference to the "signals" being processed can be especially useful.

2. Information Gathering And Post-Solution Steps. Two further possibilities for limiting the scope of a claim to an algorithm are the addition of information gathering steps and post-solution activities. However, "[g]iven that the method of solving a mathematical equation may not be the subject of patent protection, it follows that the addition of the old and necessary antecedent steps of establishing values for the variables in the equation cannot convert the unpatentable method to patentable subject matter."

Even if the steps antecedent to solution of the algorithm are new, because the algorithm is newly discovered, the subject matter is not made statutory by their inclusion in the claims.

The same is true of conventional post-solution activity. "The question in this case is whether the identification of a limited category of useful, though conventional, post-solution applications of such a formula makes respondent's method eligible for patent protection." "The notion that post-solution activity, no matter how conventional or obvious in itself, can transform an unpatentable principle into a patentable process exalts form over substance."

The CCPA held in *Walter* that, "[I]f the end-product of a claimed invention is a pure number . . . the invention is non-statutory regardless of any post-solution activity which makes it available for use by a person or machine for other purposes. If, however, the claimed invention produces a physical thing . . . , the fact that it is represented in numerical form does not render the claim nonstatutory." The court continued that “[i]f [Section] 101 could be satisfied by the mere recordation of the results of a non-statutory process on some record medium, even the most unskilled patent draftsman could provide for such a step, thus converting a nonstatutory process to a statutory one

208. *In re Christensen*, 478 F.2d 1392, 1394 (C.C.P.A. 1973). See also *In re Grams*, 888 F.2d 835, 839 (Fed Cir. 1989); *In re Warmerdam*, 33 F.3d. 1354, 1360 (Fed. Cir. 1996).


211. Id. at 590.

212. 618 F.2d at 767-68
with relative ease."213

In Flook the calculation of a new alarm limit did not make the claims statutory, but in In re Johnson,214 the Court found that the recording of new, noiseless seismic traces on a record medium caused the claims to overcome the prohibitions of Section 101. This indicates that language in a claim directed to the recording of a trace may not make the claim statutory, but the recitation of new method steps for generating trace signals may cause the claim to define statutory subject matter.

The process claimed in In re Taner,215 passed muster as statutory subject matter in part because it involved imparting seismic waves to the ground and retrieving them, as well as analyzing the results.

These decisions evince two useful claim drafting tips. First, the patent practitioner should include descriptive labels to emphasize the non-abstract significance of mathematical or numerical quantities that appear as part of a claim, e.g., "forming (or storing, comparing, or receiving) a signal having a value indicating the loudness of a sound (or pressure)." Second, the patent practitioner should incorporate the algorithm steps (or elements) in a cooperative way with the data gathering steps and hardware elements, e.g., "in a system for detecting narrow-band signals in a noisy environment, forming a set of fourier transform coefficient signals based on each of a plurality of samples from an antenna."

3. Use Versus Calculation Limitations. The Federal Circuit and its predecessor, the CCPA, have held that claims containing an algorithm are not statutory where they are "directed to a calculation and would have preempted use of the algorithm in making the calculation." However, claims are statutory "where the claims [are] to methods of operating [a] . . . system."216 "The decisive factor is whether a claimed method is essentially a mathematical calculation."217 Thus if the claims are merely directed to calculation, they are non-statutory. However, when the claims use the calculation to control some other process or apparatus, they are statutory.

The drafting tip from the above discussion is to avoid unnecessary use of mathematical names for means or steps in-

213. Id. at 770.
cluded in claims. Use nouns (e.g., "comparing means") or ger-
unds (e.g., step of "comparing") that have real world physical
significance whenever possible.

In summary, under the Freeman-Walter-Abele test, if claims
recite an algorithm covering computer program-related subject
matter, but that algorithm is not a mathematical algorithm,
they comply with 35 U.S.C. § 101. If the claims recite a mathe-
matical algorithm, they may still qualify as patentable subject
matter if the algorithm is applied to physical elements or pro-
cess steps in such a way that knowledge of the algorithm is be-
ing used, but the mathematical algorithm itself is not pre-
empted. This use of the algorithm must include more than a
field of use limitation in the claim preamble, information gath-
ering steps or non-essential post solution activity.

C. Putting Alappat, Warmerdam, and Lowry to Practice

It seems that the USPTO is making two types of Section
101 rejections. The first deals with software per se, and the sec-
ond deals with computer program-related inventions which
make use of a mathematical algorithm. Robert Greene Sterne et
al. offer specific advice, "by originally drafting the claims with
the question 'What did the applicant invent?' in mind, the
probability of receiving a statutory subject matter rejection can
be decreased."218

1. Drafting The Claimed Invention To Be Read "As a
Whole." The USPTO considers the claims of any computer pro-
gram-related invention in which the process steps or apparatus
elements are described at least in part in terms of mathematical
algorithms, "as a whole" as outlined in Diehr.219 Because of this,
the preamble and the body of the claims should be drafted with
functional language. This will serve to enable the examiner to
understand the claimed invention "as a whole." Nonetheless,
"field of use limitations," "insignificant post solution activity" or
mere "data gathering" must be carried to the specification; they
cannot render a claim directed to non-statutory subject matter
statutory by simply reciting the above concepts in the claim.
The specification for any computer program-related invention
should be drafted with statutory subject matter in mind. Also,

218. Robert Greene Sterne, et al., Preparing and Prosecuting Electronic and Com-
puter Related Applications: Avoiding and Overcoming Statutory Subject Matter Rejec-
the descriptive use of hardware is useful. Include as much hardware as possible when describing the invention, even where it consists of only software. Such a high level illustration of the computer platform on which the software operates should be described and illustrated in the patent application.

The patent practitioner should avoid only claiming the mathematical aspects of the invention in the specification. This is accomplished by explaining any transformation that is taking place, be it of signals or physical elements, or by explaining how the mathematical algorithm is applied in any manner to physical elements or process steps. It is always easier to argue that the claims as a whole satisfy the requirements of Section 101 if the invention is described in the specification at a high level.

2. Specific Claim Language Essential to Claim Drafting. The invention cannot be described simply as a mathematical algorithm. It is essential that the computer program-related method be described and claimed within the context of a machine environment, e.g., a computer. Sterne et al. offer two recommendations to accomplish this. First, it is recommended "that the words ‘computer-based’ be inserted before the word ‘method’ in method claims." They also recommend "inserting the words ‘computer-based’ before the words ‘system’ and ‘apparatus’ in apparatus and system claims." This is especially important when the system or apparatus claims are written in means plus function format.

3. Types of Claims To Be Used in Claim Drafting. To facilitate the Section 101 evaluation, the patent practitioner should use system and/or apparatus claims in the application. Sterne et al. recommend that two sets of system and/or apparatus claims be drafted. The first set should use the means plus function form under 35 U.S.C. § 112, ¶ 6. The second set should use specific architecture and/or hardware components.

220. See supra text accompanying note 195-197.
221. Sterne, supra note 218 at 310.
222. Id.
223. Id. at 311.
224. See, Ex parte Akamatsu, 22 U.S.P.Q.2d (BNA) 1915 (B.P.A.I. 1992) (holding that means plus function claims that differ from method claims only in "means for" terms before steps must be treated as indistinguishable from method claims in determining whether method is statutory subject matter).
225. See supra text accompanying notes 123-128 discussing Iwahashi. Iwahashi’s claim reciting a read only memory (ROM) was held to be statutory since the ROM was considered to be "a specific piece of apparatus," such that the claim did not wholly pre-
The patent practitioner should also include in the specification any possible hardware components that could be substituted for the computer program-related methods. Even if not claiming the specific hardware, this recitation of hardware in the specification, and especially in the claims, facilitates the Section 101 determination. Once claims are rejected under Section 101, hardware elements that are not specifically discussed in the specification cannot be added ex post facto to make the claims statutory.

CONCLUSION

The Federal Circuit decisions in Alappat, Warmerdam, and Lowry acknowledged the current practice of patent practitioners in claiming the invention as an application of the algorithm directed to a specific device or apparatus, as opposed to the algorithm itself. If a claim embraces a purely mathematical algorithm and is drawn to that algorithm, it is clearly unpatentable subject matter. If, however, the claims embracing an algorithm can be tied to the means of carrying out the functions performed by the program by describing the computer program-related method in the context of a machine or computer environment, it is patentable subject matter.

These Federal Circuit decisions are a continued expression of the Supreme Court's liberal views toward patentable subject matter and algorithms in Diehr. By allowing algorithms to be patented if they are reduced to some type of practical application, the Federal Circuit has moved away from attempting to define what a mathematical algorithm is to a more practical and useful endeavor of determining whether the claimed invention fits into one of the statutory subject matter categories of 35 U.S.C. § 101. As such, the USPTO and the courts should expand patent protection to algorithms and other computer program-related inventions on a broader scale in order to keep up with technological advances so that these new and useful inventions can be afforded patent protection. Since it seems likely that Congress will not be providing any direction in this matter, the USPTO and the courts must continue to show uniformity in their treatment of algorithm and computer program-related patent claims under the guide of Diehr.

empt the use of the algorithm recited in the claim. 888 F.2d at 1375.