
Virginia Johnson

Follow this and additional works at: https://repository.law.uic.edu/jitpl

Part of the Computer Law Commons, Intellectual Property Law Commons, Internet Law Commons, Privacy Law Commons, and the Science and Technology Law Commons

Recommended Citation

https://repository.law.uic.edu/jitpl/vol5/iss2/6

This Comments is brought to you for free and open access by UIC Law Open Access Repository. It has been accepted for inclusion in The John Marshall Journal of Information Technology & Privacy Law by an authorized administrator of UIC Law Open Access Repository. For more information, please contact repository@jmls.edu.
COPYRIGHT PROTECTION FOR
COMPUTER FLOW LOGIC AND
ALGORITHMS

The demand for computer software\(^1\) has increased rapidly within the last few years. Between 1982 and 1983 software industry revenues nearly doubled.\(^2\) The growth rate of the software industry far exceeds that of the hardware industry\(^3\) and, due to decreasing hardware costs,\(^4\) the future profits of computer technology may well lie in software. This growth has created an acute need for adequate

1. Although computer industry jargon is imprecise, the term "software" commonly refers to all the non-hardware accoutrements of computer processing. These include computer programs, flow charts diagraming program logic and algorithms, file layouts, input and output formats and documentation manuals. The term "computer program" refers to the set of instructions which tell the computer what to do. The programs are based on the logic flow charts, file layouts and input and output formats for that particular program's design. See W. CHOW & C. SIFPEL, COMPUTER GLOSSARY FOR ENGINEERS AND SCIENTISTS (1972). For a judicial definition of software, see, e.g., Honeywell, Inc. v. Lithonia Lighting, Inc., 317 F. Supp. 406, 408 (N.D. Ga. 1970). This Note will refer to the copyright law definition of the term "computer program" as follows: "A set of statements or instructions to be used directly in a computer to bring about a certain result." 17 U.S.C. § 101 (1982).

2. The revenue from sales of software increased from $12 billion in 1982 to $19 billion in 1983, a rise of 42% in one year. STANDARD AND POORS, Computer Services Industry, STANDARD AND POORS IND. SURVEYS (1984). Most of this increase reflects the demand for software caused by the advent of the desk-top, personal computer. Desk top computers have increasingly become a standard business tool as well. A 1983 survey of more than 2,000 of the largest business organizations in the manufacturing, financial and transportation industries reports that the average monthly rental cost of desk-top computers used by executives is $50,000. The survey also reports that the major factor in the selection of the hardware was the availability of compatible software. Data Decisions, Micros at Big Firms: A Survey, Datamation, Nov. 1983, at 161.

3. Revenue from hardware sales increased by 16% in 1983 compared to a 42% increase in software sales. STANDARD AND POORS, Computer Hardware Industry, STANDARD AND POORS IND. SURVEYS (1984).

4. This decrease is reflected in the fact that although software costs have remained relatively steady, they now account for 25 cents of every dollar spent on combined hardware and software packages, whereas in 1979 they accounted for only 10 cents of every dollar spent. Software for the Masses, TIME, Oct. 12, 1982. One commentator has noted that central processing units which cost millions of dollars in the 1950s are now available for under $10.00. Maggs, Some Problems of Legal Protection of Programs for Microcomputer Control Systems, 1979 U. ILL. L.F. 453.
legal protection for computer programs. The three traditional forms of intellectual property protection—patent law, trade secret law and copyright law—do not meet the industry's need. Of these three, copyright is the least often used because the scope of its protection is uncertain. Copyright law may leave flow logic and algorithms unprotected, yet they are frequently as valuable as its written code.

The uncertainty surrounding copyright is costly and unfortunate. It is costly because other, more expensive methods must be used to protect software. It is unfortunate because copyright, as a legal mechanism, appears to be flexible and simple enough to provide ample protection for what may be the most valuable part of a program. This Note attempts to limit this uncertainty by addressing the questions of whether, and to what extent, copyright law protects program logic and algorithms.

Part I identifies the major problems with the use of patent, trade

---

5. Accord, Pfeifer, Legal Protection of Computer Software: An Update, 5 Orange County B.J. 226 (1978) (stating that the need to find an adequate means of protection for computer programs is "one of the greatest challenges to emerge in recent years").

6. A proliferation of trade secret cases involving computer programs reflects the frequent use of trade secret law. See infra note 12. The desire to use patent law is reflected both in the large number of patent cases and in the fact that the Patent Office recently published guidelines for the examination of program patents. See infra note 14. By contrast, to date there have been fewer than ten cases involving copyright infringement of programs. Further, it is estimated that over one million programs have been written each year for the last decade, but only 1042 were registered at the Copyright Office as late as 1979. For a discussion of the industry's reluctance to use copyright law, see Milgrim, Protection and Licensing of Software, USC Computer Law Institute § 1 (1982). See also Green, The Computer Program Problem, Indus. Research and Dev., Dec. 1981, at 31.

7. Copyright law protects an author's expression, which means that a verbatim copy of a program code would infringe a copyright because the code is clearly the programmer's expression. See infra note 14. However, copyright also protects a level of abstract work on the theory that certain non-literal aspects of the work are part of the author's expression. See infra text accompanying notes 86-95. For a discussion of comprehensive nonliteral similarity, see 3 M. Nimmer, Nimmer on Copyright § 13.03 [A] [1] at 13-19 (1984). Characterizing computer programs as a new form of expression presents the question of what level of similarity short of verbatim copying will constitute infringement. Of the cases to date dealing with copyright infringement of computer programs, only one involved non-duplicative copying, and it was disposed of procedurally. S & H Computer Syst., Inc. v. SAS Inst., Inc., 568 F. Supp. 416, 423 (M.D. Tenn. 1983) (denial of one count of motion for summary judgment because defendant answered that he had only copied the ideas in plaintiff's programs, whether similarities were of idea or expression was a question of fact).

8. Flow logic is described in Part II of this Note.

9. Algorithms are described in Part II of this Note.

10. The significance of flow logic and algorithms is discussed in Part II.
secret and copyright law to protect programs. It concludes that, in circumstances when copyright law is adequate, its use should be encouraged because it is advantageous to society. Part II illustrates the programmer's need to protect logic and algorithms and examines a copyright statute which may preclude protection. Part III proposes an argument for copyright protection of flow logic based on its functional similarities to other forms of expression. Finally, Part IV of this Note advances an argument that compilations and combinations of algorithms may also be protected as expression.

I. PROBLEMS IN USING THE TRADITIONAL FORMS OF PROTECTION

Three forms of legal protection are available for computer programs: patent law\(^{11}\) trade secret law\(^{12}\) and copyright law.\(^{13}\) Each

\(^{11}\) Patent law, which is exclusively federal, protects the invention or discovery of "any new and useful process, machine, manufacture or composition of matter. . . ." 35 U.S.C. § 101 (1982). To qualify, the subject matter must be useful, id.; novel, or not previously known, 35 U.S.C. § 102 (1982); and non-obvious, 35 U.S.C. § 103 (1982) (the difference between the claimed invention and similar prior inventions must not have been obvious to one versed in the art). Patents are obtained by application to the U.S. Patent Office and by providing in the application a description of the subject matter and precise statements or "claims" of the subject matter sought to be patented. 35 U.S.C. § 112 (1982). If the patent is granted, the patentee has the right to exclude others from independent development of the subject matter for seventeen years. 35 U.S.C. § 154 (1982).

\(^{12}\) Trade secret law is state law which protects trade secrets from misappropriation. Although it varies from state to state, most states accept the definition of trade secrets as "any confidential formula, pattern, device or compilation of information which is used in one's business and which gives an advantage over competitors who do not know or use it." Restatement of Torts § 757 comment b (1939). Subject matter protected as trade secrets include: features of computer hardware, Telex Corp. v. IBM Corp., 367 F. Supp. 258, 320 (N.D. Okla. 1973), modified, 510 F.2d 894 (10th Cir. 1975), cert. dismissed, 423 U.S. 802 (1975); magnetic memory cores, Sperry Rand Corp. v. Petronix, Inc., 311 F. Supp. 910 (E.D. Pa. 1970); and a particular combination of programming techniques, Com-Share, Inc. v. Computer Complex, Inc., 338 F. Supp. 1229 (E.D. Mich. 1971). The subject matter must be novel, unique or must otherwise give a competitive advantage. Commonly known information cannot qualify as a trade secret. See, e.g., Wilson Certified Foods, Inc. v. Fairbury Food Prods., Inc., 370 F. Supp. 1081 (D. Neb. 1974) (no trade secret in a process for creating bacon bits since the process was widely known in the food industry). However, a unique combination of known concepts or a unique application of known concepts that gives a competitive advantage qualifies as a trade secret. See, e.g., Telex Corp. v. IBM Corp., 367 F. Supp. 258 (N.D. Okla. 1973) (hardware features which were commonly known could in combination constitute a trade secret, since they allowed IBM to achieve its performance goals of the 38309 model). The subject matter must be kept secret, thus, one must obtain explicit or implicit promises of confidentiality from those having access to the secret. See, e.g., Sinclair v. Aquarius Elec., Inc., 42 Cal. App. 3d 216, 116 Cal. Rptr. 654 (1974) (unprotected disclosure of trade secrets forfeits protection). For a general dis-
form has developed as the industry has grown, but none afford

cussion of the secrecy requirement, see R. Milgrim, Trade Secrets, § 2.07(2) (1978). Secrecy is lost by disclosure or by failure to take reasonable precautions to avoid disclosure. See, e.g., Motorola, Inc. v. Fairchild Camera & Instrument Corp., 366 F. Supp. 1173, 1186 (D. Ariz. 1973) (trade secret lost when company gave unrestricted guided tours through plant).

13. Copyright law, which is exclusively federal, protects any "original work of authorship fixed in any tangible medium of expression. . . ." 17 U.S.C. § 102(a) (1982). However, protection is limited to the author's expression in the work. It does not extend to the idea or methods explained in the work. 17 U.S.C. § 102(b) (1982); Mazer v. Stein, 347 U.S. 201 (1954) (copyright protection subsists only in the author's expression, not in his ideas). Expression extends beyond the written words, including, for example, the sequence of events. See, e.g., Sheldon v. Metro-Goldwyn Pictures Corp., 81 F.2d 49 (2d Cir. 1936) (close parallelism of events constitutes copying). Works are automatically protected from the moment they are written, recorded, filmed or otherwise fixed in a tangible medium. 17 U.S.C. § 102(a) (1982). Protection prevents another from copying, paraphrasing or otherwise creating a derivative work from the expression. 17 U.S.C. § 106 (1982). A derivative work is defined as "a work based upon one or more pre-existing works, such as a translation . . . abridgement, condensation, or any other forms in which a work may be recast or adapted." 17 U.S.C. § 101 (1982). Protection lasts for fifty years. 17 U.S.C. § 123 (1982).

14. Patent has been used to protect computer programs which are an integral part of an otherwise patentable machine or matter-transforming process. Diamond v. Diehr, 450 U.S. 175 (1981) (program which implemented new elements in the process of curing rubber was patentable as part of that process); Diamond v. Bradley, 450 U.S. 381 (1981) (software which accessed otherwise inaccessible scratch pad registers in the central processing units of certain computers was patentable because it applied to a specific physical environment); In re Abele, 684 F.2d 902 (C.C.P.A. 1982) (program method of calculating data and displaying it in pictorial form held patentable as an integral part of image-enhancement technique in a CAT scanning environment); In re Pardo, 684 F.2d 912 (C.C.P.A. 1982) (software which compiled computer programs and automatically rearranged formulas within the programs held patentable because it related to the internal operations of a particular computer.

ideal protection. Each has serious limitations as to the protection it provides.

Patent law protects only those programs which are integral parts of inventions of matter-transforming processes. For example, patent law can protect programs which control a rubber curing process, or which adjust the image on a cathode ray tube. This limitation on the scope of protection afforded by patent law excludes from protection the large number of programs which simply calculate, keep records or otherwise manage information. Several other restrictions further reduce the pool of programs eligible for protection under patent law. To receive protection, the program must be novel and non-obvious. Few programs meet this requirement because programs are often created to relieve humans of more tedious, rote and menial tasks. Moreover, the program must not only be novel, but it also must be part of an otherwise patentable process or machine. Finally, if the program contains a mathematical algorithm, the algorithm must interface with the process so that a patent does not preempt the use of the algorithm in other contexts.

15. A program which simply calculates, updates, files or retrieves information is not patentable. Gottschalk v. Benson, 409 U.S. 63 (1972) (computer program method of transforming numbers from one number system to another not patentable because not tied to a specific application involving a machine or process). But if the program automatically applies the calculations as a step in the process, the program may be specified as one of the claims in the process. Diamond v. Diehr, 450 U.S. 175 (1981) (program which automatically adjusted temperatures on the basis of its calculations held patentable).


18. Computers are most commonly used to store and retrieve information, process words, calculate and make logical predictions. This fact is reflected by the requirements of the primary fields in which computers are used: federal, state and local governments, business, finance, education, medicine, aviation, meteorology and scientific research.

19. 35 U.S.C. §§ 102, 103 (1982). Although it is impossible to say with certainty what percentage of programs will fail to meet these requirements, commentators agree that few programs will. See, e.g., Davidson, Protecting Computer Software: A Comprehensive Analysis, 23 JURIMETRICS J. 339, 357 (1983) (less than one percent will be novel and non-obvious); Stern, ROMS in Search of a Remedy: Can They Find It?, 1 COMPUTER L. REP. 1 (1981) (program improvements will seldom qualify for the non-obviousness requirement); Comment, Patentability of Computer Software: The Non-obviousness Issue, 62 IOWA L. REV. 615 (1976) (non-obviousness requirement poses a significant obstacle).


21. Parker v. Flook, 437 U.S. 584 (1978) (computer program used to monitor and time the distillation process for petroleum held not patentable because the process it implemented was not in itself novel).

22. Courts have held that if the computer claim recites a mathematical algorithm,
Even when these requirements are met, acquiring a patent may be impractical due to the time required in the application process and the relatively short lifespan of most programs. The Patent Office application process takes an average of two years, not including the time it takes a skilled attorney to draft the patent application. Yet, the commercial lifespan of most programs is less than three years.

Another drawback of using patents to protect programs is that even if granted, a patent's validity is unreliable. Over seventy percent of the patents challenged at the appellate level are held invalid, as are eighty-six percent of those considered by the Supreme Court. Presumably, patents on programs would be even less reliable given that applicability of patent law to computers is still evolving. A final drawback is that the requirement that a full description of the subject matter be available for public inspection gives competitors opportunity to misappropriate the program. Unlike machines and manufacturing processes, program methods are it must meet a further requirement in terms of its interface with the physical environment. The algorithm itself must directly transform some physical element; it cannot simply provide calculations, even though other parts of the program may automatically apply them to the physical environment. In re Walter, 618 F.2d 758 (C.C.P.A. 1981) (although programs were tied to the specific end use of seismic surveying, the claims, which recited mathematical algorithms, were not patentable because the subject matter of the claims was simply improved methodology for calculations, not improved physical products). This additional requirement reflects the principle that patent law does not preempt the use of mathematical formula.


24. Drafting claims could easily take days or even weeks because programs can be long and complex. Amicus Curiae Brief of Applied Data Research at 3, 4 & 7, Parker v. Flook, 437 U.S. 584 (1978).

25. Bender, supra note 23. But see Frank, The New Software Economics, Part 3, COMPUTERWORLD, Jan. 22, 1979, following p. 52 (because of advancing technology many programs may have a lifespan of as long as fifteen years).

26. Stern, supra note 19, at 8.

27. As the abundance of commentary on the subject reflects, many issues are unresolved in the application of patent law to programs. A major issue is that of when an algorithm is mathematical. Is an algorithm mathematical when it solves a problem that could have been solved by mathematics, or only when it recites a mathematical formula? See Novick & Wallenstein, The Algorithm and Computer Software Patentability: A Scientific View of a Legal Program, 7 Rutgers J. COMPUTERS, TECH. & L. 313 (1980). Another issue is whether the prohibition against preempting the use of mathematical algorithms should extend to the symbolic logic because all logic is natural law. See Davidson, supra note 19, at 353-55. For a history of decisions on the patentability of software, see Chandler, Proprietary Protection of Computer Software, 11 Balt. L. Rev. 195, 230-55 (1982).

not visible, and could be easily misappropriated without discovery.29

Using trade secret law for protection is expensive.30 The requirement that all those who have access to the subject matter agree to keep it secret necessitates elaborate non-disclosure agreements with customers31 and employees.32 Physical precautions such as using locked working and storage areas, passwords and sign-in and out procedures33 are necessary to secure the subject matter from inadvertent or deliberate misappropriation.

Another particularly acute problem, given the computer industry's high employee mobility rate,34 is preserving the secret when an employee leaves the company. Once an employee reveals the secret on a new job, protection is lost.35 There is no recourse against the new employer for using the secret, unless the employer knew initially that the subject matter was secret.36 Practically speaking, an

30. Commentators generally agree that the cost of the measures needed to establish and maintain the secret is the main disadvantage of trade secret protection. See Davidson, supra note 19, at 397; Gemignani, supra note 29, at 309; Pope & Pope, Protection of Proprietary Interests in Computer Software, 30 ALA. L. REV. 527, 533 (1979).
33. Telex Corp. v. IBM Corp. illustrates the lengths to which owners of software must go to protect trade secrets. In Telex, IBM presented evidence of magnetic locks on doors, guards, and television cameras, but was successful in demonstrating that trade secrets had been maintained only after showing that the program had been translated into unintelligible code. 367 F. Supp. 258 (N.D. Okla. 1973), modified, 510 F.2d 894 (10th Cir. 1975), cert. dismissed, 423 U.S. 802 (1975).
34. The shortage of trained personnel leads to frequent employee-raiding, where programmers are induced to switch companies by offers of higher salaries and greater benefits. It is estimated that the typical Silicon Valley computer analyst or programmer has worked for an average of three different employers by the time she has practiced in the profession for ten years. Missing Computer Software, Bus. Wk., Sept. 1, 1980, at 46.
36. By-Buk Co. v. Printed Cellophane Tape Co., 63 Cal. App. 2d 157, 329 P.2d 147 (1958) (former employee of plaintiff and his new employer held liable for misappro-
employee on a new job will use as much previously acquired information as possible, given the public policy against post-employment restrictions.\(^3\) Information that might be considered a trade secret outside the context of an employer/employee relationship is often found to be knowledge generically necessary to the practice of the employee's chosen business.\(^3\) Some jurisdictions even impose a higher burden of proof on an employer when an employee is the defendant.\(^3\)

Another problem with the use of trade secret law to protect programs is the possibility that the law will not protect a growing segment of the market—packaged programs which are mass distributed.\(^4\) Arguably, any claim that the subject matter has been kept secret is negated by mass distribution and the absence of non-disclosure agreements.\(^4\) The industry has tried to circumvent this
problem by printing on the package that the consumer agrees to nondisclosure by breaking the seal. The courts, however, are unlikely to enforce such an adhesive term.\textsuperscript{42} Further, even if such terms are legally enforceable, a program producer could not realistically sue thousands of consumers. Therefore, the threat of suit would not be a deterrent.

Unlike trade secret and patent law, copyright law is limited because its protection does not extend to all aspects of a program.\textsuperscript{43} The programmer would like to protect the functions performed by the program, because those functions are the essence of the program's marketability. The programmer would also like to protect novel ideas, programming techniques, and user-oriented features of the program's internal design.\textsuperscript{44} Because copyright only protects expression and not ideas,\textsuperscript{45} it will not protect these aspects of a program. Instead it only protects the form in which the program is written, including the code\textsuperscript{46} and combination of algorithms.\textsuperscript{47}

Where the most valuable aspects of a particular program are its functions or the techniques in its internal design, copyright is clearly inadequate protection. Frequently, however, the time and effort involved in developing the logic will be more valuable than the
customers without eliciting promises of confidentiality. See Davidson, \textit{supra} note 19, at 397. See also Gilburne \& Johnston, \textit{supra} note 32, at 229-32.

\textsuperscript{42} See Davidson, \textit{supra} note 19, at 396; Gilburne \& Johnston, \textit{supra} note 32, at 228-29. Both authors argue that the licensing restrictions printed on the packaging of mass distributed software will not be enforced because they are unilaterally imposed by the software producer, leaving the consumer without an opportunity to negotiate.

\textsuperscript{43} The broad definition of trade secrets given in the commonly used \textit{Restatement of Torts}, \textit{supra} note 12, encompasses the functions and internal design of a program. Although patent law does not protect mental processes, it does protect ideas as integrated with a machine or process. See \textit{supra} note 14. By contrast, copyright only protects expression—the manner in which the program was written.

\textsuperscript{44} See, Davidson, \textit{supra} note 19, at 398-400. Davidson discusses the advantages of trade secret over copyright, indicating that only trade secret law protects program functions, programming techniques and special user-oriented features. An example of a user-oriented feature that the industry would like protected, but which copyright does not protect, is the input formats used with a program. Input formats are the items of data to be entered into the computer and the order in which they are to be entered. Much effort is spent in determining the best order for input and in teaching that order to the user. A competitor who can utilize familiar formats in his program has a distinct marketing advantage. Synercom Tech., Inc. v. University Computing Co., 462 F. Supp. 1003 (N.D. Ill. 1978) (suit for damages because defendant's program used same complicated input formats as plaintiff's which gave him competitive advantage; formats held not protectable by copyright because they were ideas).

\textsuperscript{45} See \textit{supra} note 13.

\textsuperscript{46} See \textit{supra} note 14.

\textsuperscript{47} Parts III and IV of this Note argue that program flow logic and combinations of algorithms should be characterized as part of the program's expression.
In these instances, copyright law should be used because arguably it provides adequate protection while offering several advantages to society. First, copyright protection costs little compared to the costs of patent and trade secret protection. Presumably, the cost savings would be passed on to the consumer. Secondly, the certainty of copyright protection eliminates the need for time-consuming litigation. Because copyright protects all original expression, regardless of the nature of the program, questions of novelty, non-obviousness and integration with machines that arise in patent law are not a problem. Also, because copyright protection does not depend on conditions surrounding the creation of the work, issues of secrecy, or the employee’s right to the subject matter, which plague trade secret law, are avoided. Finally, copyright law is less inhibitory than trade secret and patent law on the exchange of information and growth of competition in the industry. Trade secret law retards the dissemination of technical expertise because it depends on secrecy. Thus, even when a programmer develops a technique she is willing to share, if it is part of a secret she

48. See Kolle, Computer Software Protection: Present Situation and Future Prospects, 13 COPYRIGHT 70 (1977) (coding is a relatively minor effort compared to logic development); see also Note, Computer Programs and Proposed Revisions on the Patent and Copyright Laws, 81 HARV. L. REV. 1550 (1968) (major effort in programming is in developing program logic). For a discussion of the important role the underlying logic plays, see B. NIBLETT, LEGAL PROTECTION OF COMPUTER PROGRAMS (1980).

49. Parts III and IV of this Note argue that the logic and algorithms should be protected as part of the author’s expression.

50. Obtaining copyright protection requires two steps, neither of which involve more than a nominal expenditure: 1) printing the copyright notice on the work, if it is to be published; and 2) registering the claim by depositing a copy of the work and a $10.00 fee with the Copyright Office. 17 U.S.C. §§ 408, 708 (1982). For computer programs, only the first and last 25 pages of the source code need be deposited. 37 C.F.R. § 202.20(c)(2)(vii) (1984). Technically, a work is protected without registration, but registration is required to bring an infringement action. 17 U.S.C. § 411 (1977). By contrast, trade secret protection requires the cost of drafting and negotiating nondisclosure agreements, and taking physical precautions. See supra notes 30-39 and accompanying text. Patent protection involves legal fees for drafting the claims, and application of minimum fees of $65.00, plus the cost of using other means of protection while waiting for the patent. 35 U.S.C. § 41 (1984).

51. See supra note 11.

52. See supra note 12. There is a great deal of uncertainty in trade secret law over the amount of information an employee may take from an old job to a new one. Copyright law avoids this uncertainty because the employment agreement can specify that any programs are to be works “made for hire” as defined in 17 U.S.C. § 101 (1984), which gives the employer the copyrights. For further discussion of the effects of this specification, see Angel & Tannenbaum, Works Made for Hire Under S22, 22 N.Y.L. SCH. L. REV. 209 (1976).

53. Gemignani, supra note 29, at 272, see also Comment, Copyright Protection for Computer Programs, 30 COPYRIGHT L SYMP. 1 (1982); Pope & Pope, supra note 30.
will probably not make the effort to separate it for others to learn, and other programmers will needlessly waste time repeating the work. Patent law may create needless monopolies by providing more protection than is economically required to encourage invention. Patent prevents not only copying, but independent development; yet because the value of many programs lies in the time and effort expended in the development of detailed logic and code, the programmer's economic investment is largely protected if others are prevented from copying that detail.  

II. THE SCOPE OF COPYRIGHT PROTECTION

Copyright unequivocally protects the program code as part of the author's expression. A duplication of the source code, or even a copy with minor variations, infringes the copyright. However, most of the effort involved in creating programs often lies in the development of flow logic and algorithms. If these are not protected as part of the author's expression, then copyright protection will be meaningless in many instances. No cases have addressed the question of whether flow logic and algorithms are protected by copyright law. The answer depends primarily on future interpretations of section 102(b) of the Copyright Act. This section describes flow

54. See Davidson, supra note 19 at 358. He explains that because software is symbolic, it is more vulnerable to copying than to reverse engineering; thus the broad, monopolistic protection of patent law is not necessary. He also warns that because software is symbolic, lawyers will have difficulty determining the scope of the patent claim, and that their typically conservative reaction to avoid infringement will give broader protection than is deserved. See also Y. BRAUNSTEIN, ECONOMICS OF PROPERTY RIGHTS AS APPLIED TO COMPUTER SOFTWARE AND DATA BASES (1979) (comparing the economic consequences of trade secret, patent and copyright and concluding that copyright has less of a negative impact on the market).

55. See supra note 14.

56. A basic principle of copyright law is that one cannot escape infringing the copyright because of immaterial variations. Nichols v. Universal Pictures Co., 45 F.2d 119, 121 (2d Cir. 1930) ("the copyright cannot be limited literally to the text, else a plagiarist would escape by immaterial variations"). The court in S & H Computer Sys., Inc. v. SAS Inst., 568 F. Supp. 416 (M.D. Tenn. 1983) stated in dicta that a mere rearrangement of existing expression in a program would infringe a copyright. See supra note 7 for a brief discussion of the case.

57. See Kolle, supra note 48, at 72 (coding is a relatively minor effort compared to logic development). See also Note, Computer Programs and Proposed Revisions of the Patent and Copyright Laws, 81 HARV. L. REV. 1550 (1968) (the major effort in programming is in developing the logic). See generally B. NIBLETTI, supra note 48, for discussion of the important role the underlying logic plays in programs.

58. See supra note 7 for discussion of cases decided to date involving copyright infringement of computer programs.
logic and algorithms, shows why they need protection, and analyzes the potential impact of section 102(b) on that protection.

A. THE NEED TO PROTECT FLOW LOGIC

Before the programmer writes any code, she develops the program flow logic, a logical sequence of steps the program will perform to accomplish its functions.59 The programmer uses this flow logic as a guide for writing the code, just as a builder might use a blueprint to construct a building. The programmer expresses the flow logic in flow chart format at a general or detailed level, depending on her needs. Figure A, which depicts a general level chart of the flow logic of a simple hypothetical program, code instructions based on that logic, and a narrative showing how they correspond, illustrates this close relationship of logic to code.

Developing flow logic normally entails more effort than writing the code because code words are chosen to implement the steps in the logic.60 Unlike other types of writing, code words are selected entirely for the functions they perform, not for their meaning or connotation. Selection is quick because computer language vocabularies are limited.61 Since the programmers' major effort and thought is in the development of the underlying logic, copyright must apply to this logic if protection is to be meaningful.62 A competitor would gain a valuable advantage if she could glean the underlying logic from someone else's program listing and merely contribute her own code. Also, if the logic remained unprotected, language translations, which can be done automatically, would not infringe the copyright because there would be no actionable

59. See R. BANKS & A. DOUPNIK, INTRODUCTION TO COMPUTER SCIENCE ch. 3 (1976).
60. Id.
61. Id. Figure A reflects the simple vocabulary of the types of language most frequently used in programs that do business tasks.
62. See Chandler, supra note 27. (If the logic is not protected, language translations would not infringe because unlike natural language translations, the form will be different.) See also Schmidt, Legal Proprietary Interests in Computer Programs, 21 JURIMETRICS J. 345 (1981) (If the logic is not protected, an expert could work from a detailed flow chart and produce a program which would accomplish the same task.).
FIGURE A
DELINQUENT ACCOUNT CHECK PROGRAM

program flow logic

100-160
open file

170-190
accept acct #

210-220
init upper, lower,

230-250
get file mid

260-270
is acct yes

300-330
no no

rety yes

350-360
no no

370-400
print delinqu

400-530
print current

500
accept keyin

550
close file

*numbers correspond to program code instructions

program code in BASIC

100 REM *** Delinquent Account Check Program ***
110 REM
120 REM -- Open customer file and determine total records --
130 OPEN #1, "custfile"
140 FIELD #1, ACCTNO 7,BAL 8,PAYR 8,PAYDY 3
150 EOF # 1; NORECS
160 REM
170 REM -- Accept customer account number --
180 INPUT "Enter customer account number:"; CUSTNO
190 REM
200 REM -- Binary search the customer file for custno --
210 LOWER = 1
220 UPPER = NORECS
230 WHILE ( LOWER = UPPER )
240 MID = FLOOR ((LOWER + UPPER)/2)
250 READ #1,MID;
260 IF ACCTNO = CUSTNO THEN GO TO 350
270 IF ACCTNO < CUSTNO THEN LOWER = MID +1 ELSE UPPER = MID -1
280 WEND
290 REM
300 REM -- CUSTNO Record not found in customer file --
310 PRINT "account number not found"
320 INPUT "Retry (y/n)?" ; ANSWER
330 IF ANSWER = "y" THEN GOTO 170 ELSE GOTO 510
350 REM -- CUSTNO record found: check account status --
360 IF BAL = 50.00 THEN GOTO 450
370 CURDATE = YEAR *365.2424 + DAY
380 PAYDATE = PAYR *365.2425 + PAYDY
390 IF (CURDATE - PAYDATE) < 60 THEN GOTO 525;
400 REM
410 REM -- Account is delinquent --
420 PRINT "delinquent account"
430 GOTO 480
450 REM -- Account is current --
460 PRINT "current account"
480 REM -- Wait for clerk to acknowledge --
490 IF NOT (KEYIN) THEN GOTO 480
500 REM
510 REM -- Close customer file and terminate --
520 CLOSE #1
530 END
FIGURE B
BINARY SEARCH ALGORITHM
Narrative and Flow Chart

This logic searches a file of records for a given record number using a method similar to that used to locate a listing in a phone directory. It compares a midpoint value to the given value. Then, if the value is higher than midpoint, it only searches records higher than midpoint in the next search. The steps in the flow chart below signify the following actions:

Step 1 puts the number in a field which will be adjusted throughout the search to represent the lowest record number to be searched. It also puts the total number of records in the file into a field which will be adjusted to represent the highest record number to be searched.

Step 2 finds the record number which is midway between the highest and lowest numbers to be searched.

Step 3 compares the given record number to the lowest record number. If the given number is higher than midpoint, step 4 is next. If the given number is lower than midpoint, step 5 is next.

Step 4 changes upper to be midpoint minus 1 and the process is repeated.

Step 5 changes lower to be midpoint plus 1 and the process is repeated.

flow chart

similarity in the translation. Natural language translations are infringements because the meaning is similar. However, program

63. See Gemignani, supra note 29, at 287. An infringer could use the logic and write in another language or in the same language and the program would still look different.
languages do not have meaning. Thus, logic provides the only similarities in program translations.

B. THE NEED TO PROTECT ALGORITHMS

Depending on the program, it may also be important to protect the algorithms in the program. An algorithm is a finite set of logical steps that constitutes a method of performing a processing task. Because it is a processing method, it can be abstracted and applied to other programs. For example, Figure B depicts the Binary Search algorithm that is used in the sample program to locate the given record number. This algorithm continually decreases the range of records to be searched by adjusting the upper or lower limits in response to the previous record selection. It is a common programming technique for locating a given record in an ordered set.

Figure B here

Most programs are comprised of numerous complex algorithms. Although the value of a program may not be lost because another programmer has adopted one or even several of the algorithms, the value of the original program would be substantially diminished if all the algorithms were abstracted and used in another program.

C. INTERPRETATION OF SECTION 102(B)

Whether copyright protects flow logic and algorithms depends on how the courts will interpret section 102(b) of the Copyright Act, which states:

In no case does copyright protection for an original work of authorship extend to any idea, procedure, process, system, method of operation, concept, principle or discovery, regardless of the form in

64. See R. Banks & A. Douppnik, supra note 59, at ch. 3.

In particular an algorithm is characterized by these properties:

(1) Application of the algorithm to a particular input set or problem description results in a fine sequence of actions.

(2) The sequence of actions has a unique initial action.

(3) Each action in the sequence has a unique successor.

(4) The sequence terminates with either a solution to the problem, or a statement that the problem is insoluble.

Encyclopedia of Computer Science (1976), at 47-48. Many legal commentators do not distinguish the program flow logic from algorithms when discussing copyright protection for programs. See, e.g., Kolle, supra note 48, at 71 (“A computer program is a material form of the algorithm.”); Breyer, The Uneasy Case for Copyright: A Study of Copyright in Books, Photocopies and Computer Programs, 84 Harv. L. Rev. 281 (1970) (“The underlying logical steps may be characterized as the program’s algorithms.”). However, the distinction does exist, as is illustrated in Figures A & B, and is important in discussing the protection that may be given by copyright law. As Part III of this Note indicates, algorithms may not be protectable because of theories which would not preclude protection for flow logic.
which it is described, explained, illustrated or embodied in such work.\textsuperscript{65}

If the courts use the terms found in the statute to define flow logic or algorithms, the statute will preclude protection.\textsuperscript{66}

The courts have relied on two sources for guidance in applying the statute to computer programs: legislative history\textsuperscript{67} and the recommendations of the National Commission on the New Technological Uses of Copyrighted Works (CONTU), which was formed to determine the desirable extent of copyright protection for computer programs.\textsuperscript{68} Neither source is very helpful.

The legislative history, combined with the tone of the statute, reflects a fear of over-protection by emphasizing an intent to protect only expression while processes or methods embodied in the program remain outside the scope of the statute.\textsuperscript{69} Unfortunately, the legislative history does not provide examples of what should be characterized as the program's method of operation, procedure or process.

The recommendations of CONTU are similarly vague in that the

\textsuperscript{65} 17 U.S.C. § 102(b) (1982).

\textsuperscript{66} Only one opinion has interpreted 17 U.S.C. § 102(b) as applied to a computer program copyright infringement action: Apple Computer Inc. v. Formula Intern, Inc., 562 F. Supp. 775 (C.D. Cal. 1983). The case dealt with duplicative copying of operating system programs. Operating system programs control the functioning of the computer. The court held that § 102(b) should not be interpreted to preclude the protection of operating system programs, even though they may be a process or method or are controlling the computer. The court reasoned that such an interpretation would unavoidably preclude protection for all programs, because all programs control machine functions. \textit{Id.} at 784.


\textsuperscript{68} CONTU was created by P.L. 93-573, enacted in December 1974. The Commission conducted numerous hearings and studies between January 1975 and February 1979 to determine whether copyright law was an appropriate form of protection for computer programs and if so, the desirable scope of protection. The Commission concluded that copyright law was a better form of protection than patent or trade secret law and that the scope of protection should be determined by the courts. \textit{FINAL REPORT OF THE NATIONAL COMMISSION ON NEW TECHNOLOGICAL USES OF COPYRIGHTED WORKS} (1978) [hereinafter cited as \textit{CONTU REPORT}].

\textsuperscript{69} The House Report states:

Some concern has been expressed lest copyright in computer programs should extend protection to the methodology or processes adopted by the programmer, rather than merely to the "writing" expressing his ideas. Section 102(b) is intended, among other things, to make clear that the expression adopted by the programmer is the copyrightable element in a computer program, and that the actual processes or methods embodied in the program are not within the scope of the copyright law.

Final Report fails to address the scope of protection in exact terms. The Commission's informal responses to testimony from the computer industry indicate that copyright law should protect the sequence of steps underlying the code as part of the author's expression. The Final Report is not as precise as the informal responses. In acknowledging the difficulty of distinguishing between copyrightable programs, the Final Report suggests that Baker v. Selden be used as a source for guiding principles in making the distinction.

Baker v. Selden, an 1879 copyright decision, established the principle now codified in section 102(b) that copyright law does not protect the method or "useful art" explained in a work. The copyrighted work in Baker consisted of an essay explaining a system of accounting. The uncopyrightable method was the accounting system explained in the work. Unfortunately, it is difficult to apply the principle used in Baker to determine the desired scope of protection for computer programs. The subject of Baker, a system of accounting, is fundamentally different from a computer program. There is no parallel in a computer program to the term "method" as it was used in Baker. In Baker, the term referred to a method (the accounting system) that the copyrighted work explained, and which was distinct from the expression in the copyrighted work itself. By contrast, a computer program does not explain a method. The logic of a program may be referred to as a "method" of manipulating the computer, but this logic is also the structure or expression of the program. Due to this distinction in subject matter, Baker cannot anticipate or be analogized to the issue of protecting expression that is also a method. In the few cases that do address the issue, protec-

---

70. See generally Contu Report, supra note 68, at 2.
71. In the Commission's meetings with members of the computer industry, the Commission frequently inquired whether the underlying steps needed protection and whether protecting them would monopolize the ideas of the program. Industry members almost unanimously replied that the logic steps needed protection, and several commissioners agreed. For example, Commissioner Arthur Miller stated, "If there's another way somebody can do it, let's protect the sequence of steps. But don't block or access or ability to use the technique. It's hard to believe copyright does not extend to the series of instructions." Commissioner Gabriel Perle noted, "We should protect how somebody sat down and figured how to fix in one form or another his idea so that idea could have some utility." Transcripts of Contu meetings, National Technical Information Services, Springfield, VA. Meetings 6-7, 10, 12-14 and 19.
73. 101 U.S. 99 (1879).
74. Id. at 102.
75. Id. at 105.
76. Professor Nimmer suggests that § 102(b) should not be used as an adoption of an expanded reading of Baker to find that there is no liability for copying a work
tion has not been vitiated merely because the expression also embodied a method.\textsuperscript{77}

\textit{Baker} also established the "incidents of the arts" rule.\textsuperscript{78} According to this principle, expression needed to employ the method explained in a work cannot be protected.\textsuperscript{79} In \textit{Baker}, the "incidents of the arts" rule was applied to an almost identical copying of ledger forms illustrated in the essay which were necessary for the use of the accounting system. However, this rule is not helpful in determining whether flow logic and algorithms are protected because the rule cannot be applied until the unprotectable method has been identified. In addition, the courts may be unwilling to apply the rule to computer programs because such an application could work to nullify all copyright protection.

If the courts characterize the particular combination of algorithms as unprotectable method, the application of the "incidents of

\textsuperscript{77} Pantone, Inc. v. A.I. Friedman, Inc., 294 F. Supp. 545 (S.D.N.Y. 1968). The copyrightable work was a color matching system which consisted of a book of color samples with key numbers matched to ink formulas used by ink manufacturers. The color system embodied a method of displaying color samples by presenting a primary color and surrounding it with light and darker shades. This was achieved by adding either white or black to the samples. The court found that the arrangement employed to facilitate selection and matching of colors was a mode of expression not previously published and found infringement when the defendant produced a book of color samples using the same arrangement of primary and shaded colors. In Trebonik v. Grossman Music Corp., 305 F. Supp. 339 (N.D. Ohio 1969), the copyrightable work was a device of paper dials which enabled the user to locate the proper finger pattern on a guitar to play a desired chord. It illustrated a method of finding the finger pattern on a guitar by juxtaposing three different chord categories. The court found that the arrangement and presentation of the chords was original and novel, and found infringement where the defendant published a book using the same chord juxtapositions. Similarly, in Nikanov v. Simon & Schuster, 246 F.2d 501 (2d Cir. 1957), the copyrightable work was a chart of letters depicting the Russian alphabet. The chart illustrated a method for teaching the alphabet. The expression of this teaching method was the categorical division of the letters into groups: (1) those with pronunciations similar to their Latin equivalents; (2) those with appearances similar to their Latin equivalents; and (3) those unrelated to Latin. The court found this arrangement to be unique and found infringement when defendants published a book using the same division of letters.

\textsuperscript{78} 101 U.S. 99, 102 (1897). After stating that copyright would not protect any useful art explained in a copyrighted work, the \textit{Baker} court stated: "And where the art it teaches cannot be used without employing the methods and diagrams used to illustrate the book or such as are similar to them, such methods and diagrams are to be considered as necessary incidents to the art, and given therewith to the public. . . ."

\textsuperscript{79} Id. at 103.

\textsuperscript{79} Id.
the arts" rule could preclude copyright protection for flow logic as well. Flow logic is often necessary to the use of a particular combination of algorithms. Similarly, judicial characterization of flow logic as unprotectable method could preclude protection for program code as well, due to the close relationship between code and logic making logic a necessary incident to use of the code. Such broad cancellation of copyright protection appears to conflict with the Legislature's intent to protect computer programs. Thus far, courts have ignored the dictates of Baker when its application effectively cancels copyright protection for an entire class of works. Similarly courts may ignore Baker in defining the scope of protection for computer programs.

Because neither the Legislature nor CONTU have provided definitive guidance as to whether or not flow logic and algorithms should be protected as expression, the courts will continue to make decisions based on comparison with other forms of expression protected by copyright.

III. FLOW LOGIC AS A FORM OF EXPRESSION

Copyright law strives to achieve two goals: (1) protection of the commercial value of an author's work; and (2) societal access to the author's ideas. To meet the first goal, courts have protected non-

80. See H. Rep. 1476, supra note 67, at 57; see also id. at 54, stating that the copyright definition of literary works "includes computer data bases and computer programs to the extent they incorporate authorship in the programmer's expression of original ideas, as distinguished from the ideas themselves."

81. See, e.g., Imperial Homes Corp. v. Lamont, 458 F.2d 895 (5th Cir. 1972). The court rejected Baker to find infringement in a copy of architectural plans, stating "We simply do not read the ambit of statutory copyright protection for such a case so narrowly." Id. at 899. See generally 3 M. Nimmer, supra note 7, at § 2.08 [D] (discussing copyright protection of architectural plans notwithstanding Baker).

82. Baker itself supplies further support for an argument that the case should not be read to preclude protection for programs. The court states that its principles are confined to situations in which the work was intended for explanation and the copier wants to use the work in a non-competitive manner. Programs are not intended for explanation; they are intended for computer use. Programs may be copied for academic purposes; the copying harming the industry is copying for competitive commercial use. Baker v. Selden, 101 U.S. 99, 105 (1897).

83. See Mazer v. Stein, 347 U.S. 201, 219 (1954). In holding that the non-utilization aspects of a utilitarian object were protected by copyright, the court stated:

The economic philosophy behind the clause empowering Congress to grant patents and copyrights is the conviction that encouragement of individual effort by personal gain is the best way to advance public welfare through the talents of authors and inventors in the Science and useful arts.

The Court also acknowledged the competing goal of ensuring the free exchange of ideas. Id. at 220. See generally, 3 M. Nimmer, supra note 7, at § 1.10 for a discussion of judicial attempts to balance the competing concerns.
literal aspects of work, such as plot incidents, by characterizing them as part of the author's expression. This section describes the similarities between flow logic and these non-literal expressions, and argues that because of these similarities, flow logic should also be characterized as expression.

A. SIMILARITIES BETWEEN FLOW LOGIC AND EXPRESSION IN NON-DRAMATIC WORKS

In non-dramatic literary works, the selection of content and its arrangement is frequently deemed to be the author's expression. The rationale for protection is that the selection and arrangement evinces originality, and it is that originality which copyright law protects. This principle protects selection and arrangement of content in diverse subject matter including particular types of fantasyland settings and characters; selection of content and topical sequence of presentation in a psychological textbook; selection of information and manner or presentation of math formulas on a piston rod size chart; and content of problems chosen for inclusion in a physics text.

Program flow logic is analogous to these types of protected expression. The analogy is most clearly seen in cases protecting a combination of content selection and teaching methods. For example, in *Nutt v. National Institute for the Improvement of Memory*, the copyrighted work was a series of lectures teaching memory techniques. The infringing work contained numerous similarities, both in the topics presented and in the methods of presentation, to that of the copyrighted work. The court found infringement on the

---

84. Sheldon v. Metro-Goldwyn Pictures Corp., 81 F.2d 49 (2d Cir. 1936) (close parallelism of plot incidents is copyright infringement).

85. See infra notes 86-95 and accompanying text; see also Chaffee, Reflections on Copyright Law, 45 Colum. L. Rev. 503, 513 (1945). In addressing the issues of ideal copyright protection, Professor Chaffee recognizes that a somewhat abstract level of expression must be protected. He states: "No doubt, the line does lie somewhere between the author's idea and the precise form in which he wrote it down. I like to say that the protection covers the 'pattern' of the work."

86. Sid & Marty Kroft T.V. Prod. v. McDonald's Corp., 562 F.2d 1157 (9th Cir. 1977).

87. Meredith Corp. v. Harper & Row Publishers, Inc., 378 F. Supp. 686 (S.D.N.Y. 1974) (a preliminary injunction was granted even though the court found some paraphrasing in the infringing work, the focus of the decision was on the combination of the structural and content similarity).


90. See supra note 77.

91. 32 F.2d 236 (2d Cir. 1929).
grounds that topic selection and treatment was the author's expression.

Similarly, program flow logic is a selection of topics and treatment methods. For example, in the sample program, the programmer has selected the tasks that the computer performs in order to achieve the goal of the program: opening the file, accessing the given record, checking for a delinquent balance and the last payment date, and displaying the results to the operator. Those topics may be treated in a variety of file handling methods, such as the decision to use binary search logic and logical methods of making comparisons (for instance, using the number of days in the century for determining last payment date) and the sequencing of the logical steps.

Arguably, flow logic should be characterized as expression because it is similarly a combination of selection and arrangement. The selection and treatment of the tasks to be performed by the computer are related to the finished program code in the same way that the selection of content and presentation methods in Nutt were related to the words spoken in the lectures. The rationale underlying Nutt and similar cases, that copyright protects the originality evinced by selection and order, applies equally to flow logic because flow logic is a product of the programmer's originality.

B. SIMILARITIES BETWEEN FLOW LOGIC AND EXPRESSION IN DRAMATIC WORKS

In dramatic works, the sequence of events is frequently protected as part of the author's expression. For example, if Shake-
speare's works were not in the public domain, West Side Story might be an infringement of Romeo and Juliet because of the similarities of events in these two plays.\textsuperscript{94} Program flow logic is analogous to a plot, or series of events in a play.\textsuperscript{95} The logic is an abstraction of the written code in the same way that the plot line is an abstraction of the play dialogue. If one were to describe a play, one would relate the incidents expressed by the dialogue. Similarly, if one were to explain a program, one would relate the logical sequence of events implemented by the code. Both logic and plot provide the underlying structure of the literal words. Program code expresses the logic flow just as dialogue expresses the plot line. Further, both flow logic and plot are coherent units. The events in a play must create a dramatic unity. Similarly, the steps in flow logic must be causally related to each other and function as a logical unit.

The analogy of flow logic to the protectable story line in dramatic works is imperfect. The story line of a dramatic work has dimensions that are not present in flow logic, such as characters and tone, aesthetic effect on the audience, and personality traits of characters. Similarities in characters and dramatic effects, as well as similarities of incidents, frequently contribute to a court's conclusion that expression has been copied.\textsuperscript{96} However, the fact that flow

---

\textsuperscript{94} When courts have denied protection to "plot," they have defined it as the equivalent of an abstract idea. But when it is more properly defined as the sequence of events by which the author expresses his theme, it constitutes a pattern which is sufficiently concrete to warrant a finding of substantial similarity.

3 M. Nimmer, supra note 7, at § 13.29. See also Sorenson & Sorenson, Reexamining the Traditional Legal Test of Literary Similarity: A Proposal for Content Analysis, 37 Cornell L. Rev. 638, 650 (1952). Professors Sorenson indicate that a similarity of plot incidents would constitute similarities of expression. For example, if eight of eleven plot situations were similar, a conclusion could be drawn as to whether the pattern of similarity was due to chance.

94. 3 M. Nimmer, supra note 7, at § 13.26, 27. In listing the thirteen events in West Side Story which he thinks would infringe Romeo and Juliet, Professor Nimmer states that the events constitute a description of the work which is in some degree abstract but is far from that level of abstraction where only the basic idea is common to both.


In an actual program, the particular sequence and choice of available algorithms, manipulations, etc., as embodied in the expression of the program also constitute valid elements of expression. They are like the incidents on the development of a story or novel, whose choice and sequence is protected by copyright.

96. See, e.g., Twentieth Century-Fox Film Corp. v. Stonesifer, 140 F.2d 579 (9th Cir. 1944) (similarities in characters as well as similarities of incidents contributed to a finding of infringement); Shipman v. R.K.O. Radio Pictures, 100 F.2d 533 (2d Cir. 1938) (dissimilarity of characters was partial bases for a finding of no substantial sim-
logic lacks these dimensions does not vitiate the analogy for the purpose of argument. Flow logic remains the functional equivalent of story line because it is the structure underlying the code in the same way that story line is the structure underlying the dialogue. Just as the courts have recognized the need to protect the underlying structure in dramatic works, they may also recognize the need to protect it in programs.

C. Flow Logic as Contrasted with Ideas and Methods

Flow logic is clearly dissimilar to what has been characterized as unprotectable ideas or methods. It is dissimilar to mere ideas in that it is developed thought, whereas mere ideas are not considered to be developed thought. The subject matter in Synercom Technology v. University Computing Co. illustrates this difference. The copyrighted work in Synercom was a data input format which simply specified the order in which data items are to be entered into the computer. The court held the format to be idea, not expression, because the mere ordering of items involved no effort or development of thought beyond the concept of ordering the items. Although flow logic involves an ordering of items, the effort involved and the finished work require effort beyond a mere ordering of items. Each step in the flow logic accomplishes a task, and the intellectual process of determining the steps to be used is greater than that of selecting and arranging data items. Also, the steps in flow logic express a logical order of progression from one point to another. Thus, the mental process of ordering the steps is more complex than the simple juxtaposition of data items. Although flow logic may be a spare form of expression, it is more similar in kind and extent of effort required to dramatic incidents, text or non-dramatic works than it is to the ordering and sequencing of items which have been deemed idea, as illustrated in Synercom.

Flow logic is also dissimilar to an unprotectable method or process in a copyrighted work. Unprotectable methods are concepts explained by the work, such as shorthand systems, insurance

---

97. See supra note 93.
99. Id. at 1014.
100. Brief English Sys., Inc. v. Owens, 48 F.2d 555 (2d Cir. 1931), cert. denied, 283 U.S. 858 (1931).
schemes and game designs, rather than methods of expression embodied in the work. Because flow logic is part of the work itself and not merely a concept that the work explains, flow logic arguably escapes the holding of Baker v. Selden that copyright does not protect methods that merely explain concepts.

D. APPLICATION OF THE ABSTRACTIONS TEST

To determine whether similarities are expressions or ideas, the courts have frequently applied the abstractions test. Judge Hand first articulated this test in Nichols v. Universal Pictures to determine whether a thematically similar movie appropriated the idea or expression of a play. The test provides that where copied material is so general that it could be expressed in a variety of ways, it is an idea. But when the copied material is detailed, and does not allow for significant further development, then it is an expression. For example, in Nichols the two works were similar in that they both in-
volved conflict caused by the marriage of offspring of hostile Jewish and Irish families and the reconciliation after the birth of a grandchild. The two works were dissimilar in a number of respects, including the occurrence of events, character personalities, and the emotions that motivated characters' actions. The court held that the similarities were general enough to be considered an idea because the basic situation was developed differently in two works in that the characters' personalities, emotions and actions varied.

Similarly, one can distinguish between an idea and an expression in a computer program by applying the abstractions test. For example, in the program depicted in Figure A, the various tasks performed by the computer program, such as opening the file, accessing a given record, checking financial information, and reporting to the user, may be considered ideas because they can be developed differently. The expression which performs those tasks is the combination of the binary research logic, the logic for checking the last payment date, and the intervals of displaying data on the screen. Thus, the abstractions test would allow another programmer to write a program that performs the same general tasks to check a given record for financial information but would not allow him to copy the detailed logic flow. Applying to programs the traditional test for distinguishing idea from expression, flow logic would be considered an expression. This supports the argument that flow logic should be characterized as an expression.

In acknowledging that meaningful copyright protection must extend to a level that is somewhat abstract, courts have protected the selection and arrangement of content and the sequences of incidents as the author's expression. Because flow logic is related to the literal program code in the same way that these expressions are related to their written words, it may be similarly protected.

IV. PROTECTION FOR ALGORITHMS

A. COPYRIGHT DOES NOT PROTECT ALGORITHMS

Algorithms are not readily comparable to either the idea or expression side of the copyright law dichotomy. Because an algorithm involves developed thought, it is more than a mere idea.107 Yet it differs from an expression because it is a technique that can be adopted and used in other programs.108 Copyright probably does

107. See Comment, Patent Protection for the Algorithms in Computer Programs, 7 RUTGERS COMPUTERS, TECH & L.J. 313 (1978). The comment provides two algorithms expressing the Pythagorean Theorem and concludes that if the idea can be expressed in two algorithms, the algorithms are not the idea itself.

108. See generally R. BANKS & A. DOUPNİK, supra note 59, at ch. 3.
not protect algorithms because the idea/expression unity principle or the provisions of 17 U.S.C. § 102(b) preclude protection.

The idea/expression unity rule provides that where an idea can be expressed in only one or a limited number of ways, the expression of that idea cannot be protected. For example, in *Herbert Rosenthal Jewelry Corp. v. Kalpakian,* the copyright on the design of a jeweled pin was not infringed by an almost identical pin because the idea of a jeweled pin could be expressed in only a limited number of ways. The court reasoned that because the idea and its expression were inseparable, protection of the expression would, in effect, monopolize the idea. Although most algorithms are more complex than the simple expression usually affected by this rule, the rule extends to algorithms because the idea of the algorithm is inseparable from its expression. For example, the idea underlying the binary search algorithm in Figure B is that when searching for a given number in an ordered set, the pool to be searched can be narrowed by adjusting the range of numbers higher or lower after each attempt. Implementation of expression of this concept requires that certain steps be performed in a certain order. Although slight variations are possible, the expression of the concept is limited by the constraints of logic.

Algorithms may also be precluded from protection as methods, processes or procedures as these terms are used in 17 U.S.C. § 102(b). Although the legislative history does not use the term algorithms, it does indicate the intention that copyright not prevent the free exchange of technological ideas. Because algorithms are processing techniques that can be and customarily are exchanged, they are probably within the ambit of what the legislature does not want to protect.

---

109. See, e.g., *Herbert Rosenthal Jewelry Corp. v. Kalpakian,* 446 F.2d 738, 742 (9th Cir. 1971) (although defendant may have copied the plaintiff's expression in the design of a jeweled pin, there was no infringement because the idea of a jeweled pin and its expression was inseparable); *Morrissey v. Proctor & Gamble Co.,* 379 F.2d 675 (1st Cir. 1967) (nearly identical written instructions for a game did not infringe because there were only a limited number of ways of expressing the simple game idea).

110. 446 F.2d 738, 742 (9th Cir. 1971).


113. See supra note 69.

114. Computer Science school libraries contain textbooks of algorithms published with the intent that they be copied. Examples of titles are: *Collected Algorithms of...*
B. PROTECTION FOR COMBINATIONS AND COMPILATIONS

As discussed previously, copyright protects content selection and arrangement of content in a copyrighted work.\textsuperscript{115} \textit{Nikanov v. Simon \& Schuster}\textsuperscript{116} provides an example of subject matter protected under this theory that is analogous to the combination of algorithms in a program. The copyrighted work was a chart used to teach the Russian alphabet. The chart was divided into several sections, each constituting a method for teaching some aspect of the language. The defendant's chart differed in some visual respects, but it included the same combination of methods. The court found an infringement, holding that the total combination of methods illustrated in the chart were the author's expression.\textsuperscript{117} Similarly, the combination of algorithms used in a particular program may be characterized as an expression.

Copyright law also protects compilations\textsuperscript{118} as original works of authorship. A compilation is defined in the Copyright Act as a work formed by the collection and assembling of pre-existing materials or of selected data.\textsuperscript{119} Catalogues,\textsuperscript{120} restaurant guides\textsuperscript{121} and annotated citations\textsuperscript{122} are examples of works protected under this theory. Protection prevents another person from copying so many of the collected items that the works are substantially similar. Arguably, a set of algorithms in a program are protectable under this theory. A program is an assembly of the algorithms selected by the programmer. The algorithms are pre-existing materials because, before they are joined in the program, the programmer has either developed them or selected them from textbooks or other programs.

Assuming individual algorithms are not protected under 17
U.S.C. § 102(b), the issue in protecting them under either theory is whether these theories apply when a statute prohibits protection on an individual basis. Two conflicting decisions, *Runge v. Lee* and *Universal Sales v. Salkeld*, indicate courts may still be willing to apply the theories to algorithms.

In *Runge v. Lee*, the infringing work was a book of facial exercises that described the same sixteen exercises as the copyrighted work. The court upheld the validity of instructions that allowed the jury to find infringement because the author did not treat the subject matter independently, but relied on the labor of the other. The books were similar only in that both included the same exercises. The Court did not mention any literal similarities. In affirming the verdict of the district court, the Third Circuit in effect extended copyright protection to a combination of methods as the author's expression.

In *Universal Sales v. Salkeld*, the infringing work was a chart depicting the exercises to use with an accompanying exercise machine. Although graphically dissimilar, the exercises were the same as those on the copyrighted chart. Conflicting with *Runge*, the court found that exercise positions were not protectable because they were ideas. Acknowledging that plaintiffs had labored to ascertain the physiologically proper positions, the court said that the defendant was nevertheless entitled to adopt the positions because they were merely similar ideas and not substantially similar expressions.

These conflicting decisions reflect the courts' willingness to forego consistency in drawing the line between expression and idea when the value of an author's work would be left completely unprotected. In *Runge*, the value of the plaintiff's work was demonstrably diminished by the defendant's book, as defendant's book was published first and publishers were unwilling to publish a second book on the subject. The defendant acted unfairly because she had learned the exercises from the plaintiff and knew the plaintiff was planning to publish them in a book. In contrast, the plaintiff in *Universal* was not substantially harmed by the similarity of the charts because the charts accompanied exercise machines and were useful only in explaining how to operate the machines. The plain-

---

123. 441 F.2d 579 (9th Cir. 1971), cert. denied, 404 U.S. 887 (1971).
125. 441 F.2d at 579.
126. *Id.* at 582.
127. 511 F.2d at 904.
128. *Id.* at 909.
129. 441 F.2d at 580-81.
tiff's commercial interest was primarily in the machines. Competing charts would not decrease the market demand for those machines. The alleged copying also lacked the element of unfairness because the defendant manufactured a similar machine and, therefore, had a bona fide need to produce instructional charts. The Runge decision emphasizes the unfair and commercially competing aspects of use. This indicates that the courts may apply the combination or compilation theories to protect the algorithms in a program when a competitor has copied them and, in so doing, has substantially diminished the program's value.

CONCLUSION

Copyright law has the potential of providing better protection against misappropriation than is currently provided by either patent law or trade secret law. In order to provide this protection, ambiguities in the Copyright Act must be resolved to extend copyright protection to the flow logic underlying the program code and, in some programs, to the particular combinations of algorithms comprising the program.

This protection would not conflict with the goals of copyright law; flow logic and algorithms are analogous to other forms of expression already protected under copyright law. Further, such protection would not necessarily impede the exchange of technological knowledge. Courts could apply the abstractions test to flow logic in ways that allow the copying of ideas. Courts could also apply the combination and compilation theories to find a compromise in protecting algorithms.

Virginia Johnson*

130. See 511 F.2d at 909.
131. See id.; Stern, supra note 19. Stern states: "Legal protection of algorithms will hamper scientific progress only if the amount of protection is so great as to have that effect." Id.

* Associate in the firm of Best, Best & Krieger, Riverside, California. This Note was written while Ms. Johnson was a student at the University of California at Davis Law School. As earlier version won first place in the ASCAP Nathan Burkan competition at U.C. Davis.