

SIX COPYRIGHT THEORIES FOR THE PROTECTION OF COMPUTER OBJECT PROGRAMS*

I. Trotter Hardy, Jr.**

In recent years, many courts have addressed the question whether a computer program, in a form ready for immediate use by a computer, is copyrightable. After some initial negative answers,¹ most courts² and commentators³ have agreed that these programs are indeed copyrightable.

A decision that computer programs in any form are protectible under the Copyright Act⁴ is intuitively satisfying. Programs, after all, take a lot of creativity and effort to create;⁵ their market value can be stolen by a simple act of copying; consequently the need for protection to serve as an incentive to creation seems at least as great for computer programs as for any other copyrightable commodity.⁶ Unfortunately, none of the court decisions that have found programs in their final, usable form to be copyrightable is consistent with the language of the Copyright Act.

Under the terms of the Act, computer programs, in a form ready for use

* Research for this article was supported by a summer research grant from the College of William and Mary.

** Assistant Professor of Law, Marshall-Wythe School of Law, College of William and Mary.

1. See *Apple Computer, Inc. v. Franklin Computer Corp.*, 545 F. Supp. 812, 817-22, 824-25 (E.D. Pa. 1982), *rev'd*, 714 F.2d 1240, 1246-49 (3d Cir. 1983), *cert. dismissed*, 104 S. Ct. 690 (1984); *Data Cash Systems, Inc. v. JS&A Group, Inc.*, 480 F. Supp. 1063, 1067-68 (N.D. Ill. 1979), *aff'd on other grounds*, 628 F.2d 1038 (7th Cir. 1980).

2. See, e.g., *Apple Computer, Inc. v. Formula Int'l, Inc.*, 725 F.2d 521, 524-25 (9th Cir. 1984); *Apple Computer, Inc. v. Franklin Computer Corp.*, 714 F.2d 1240, 1246-49 (3d Cir. 1983), *cert. dismissed*, 104 S. Ct. 690 (1984); *Williams Elecs., Inc. v. Artic Int'l, Inc.*, 685 F.2d 870, 876-77 (3d Cir. 1982); *Midway Mfg. Co. v. Strohon*, 564 F. Supp. 741, 750-52 (N.D. Ill. 1983).

3. See, e.g., Note, *Copyright Protection of Computer Program Object Code*, 96 HARV. L. REV. 1723, 1743-44 (1983); Note, *Copyright: Computer Firmware: Is It Copyrightable?*, 36 OKLA. L. REV. 119, 134-35 (1983). But see Stern, *Another Look at Copyright Protection of Software: Did the 1980 Act Do Anything for Object Code?*, 3 COMPUTER L.J. 1, 16-17 (1981).

4. 17 U.S.C. § 101-810 (1976).

5. For example, *Apple Computer, Inc.*, estimated that certain of its computer programs, copied by a competitor, "took 46 man-months to produce at a cost of over \$740,000, not including the time or cost of creating or acquiring earlier versions of the programs or the expense of marketing the programs." *Apple Computer, Inc. v. Franklin Computer Corp.*, 714 F.2d 1240, 1245 (3d Cir. 1983), *cert. dismissed*, 104 S. Ct. 690 (1984).

6. *Midway Mfg. Co. v. Strohon*, 564 F. Supp. 741, 751 (N.D. Ill. 1983).

by a computer, cannot sustain an independent copyright as literary works because they are not original works of authorship. For the same reason, they cannot sustain copyright protection as derivative works. Because they do not reflect the merger of two expressions, they cannot sustain protection as joint works either. Though not independently copyrightable, these programs may yet be protectible under several related copyright theories. First, authors may rely on their right to control the preparation of derivative works to prevent the reproduction of their programs. Second, these programs might simply be "copies" of earlier, copyrightable forms of the same programs. Third, authors may rely on the "pattern" of the earlier forms of programs to control the copying of that pattern in the forms ready for use by a computer.

To follow these theories, one needs to understand that computer programs can exist in different forms. Many explanations of programs have appeared in the legal literature already,⁷ but for the reader who is unfamiliar with them, a short explanation is in order.

FORMS OF COMPUTER PROGRAMS

When programmers write programs, they write in one or more programming languages or perhaps in some form of notational system or flow chart that will later be turned into expressions in a programming language. The resulting program, written on paper or displayed on a video display terminal, is readable by the programmer, but not directly usable by the computer. Before the program can be used or "run" by the computer, it must be translated into a more detailed series of instructions, expressed in a kind of code consisting of the digits one and zero.

The analogy of a player piano roll may be helpful here. A composer may write music to be performed by a player piano, but the composer will not likely compose by punching holes in the piano roll. Rather, he will write musical notation on music paper. Someone or something must then translate that notation into the holes in the piano roll before the piano can play the composition. The translation from musical notation to holes on a roll is similar to the translation of a programmer's program, as it was written, into a version that the computer can "play." Computer experts refer to the version written by the programmer as the "source program;" the translated version, ready for the computer to run, is called the "object program."

One breakdown in this analogy is that a musical translation can be reversed: an expert given a piano roll could, with patience and effort, reproduce the sheet music for the work. Most computer program translations from source to object form are not reversible: an expert given an object program produced from a source program language, like Basic or Cobol, cannot reproduce the original source program. The program's functions—its processes, its logic, its sequence of programmed events—could be repro-

7. See discussions cited *supra* note 3. See also *Paine, Webber, Jackson & Curtis, Inc. v. Merrill Lynch, Pierce, Fenner & Smith, Inc.*, 564 F. Supp. 1358, 1366-67 (D. Del. 1983); *Apple Computer, Inc. v. Franklin Computer Corp.*, 545 F. Supp. 812, 812-16 (E.D. Pa. 1982), *rev'd*, 714 F.2d 1240 (3d Cir. 1983), *cert. dismissed*, 104 S. Ct. 690 (1984).

duced in a source program language like Basic or Cobol, but there would be no way of knowing how close such a program's phrasing or word choice would be to the original source program.⁸

In the sense that object programs cannot be translated back to their source program form, programs are much like instructions of any kind. One might, for example, ask a mason to build a brick wall of a certain height, length, width and pattern. An experienced mason would be able to follow these directions, but someone with no knowledge of or experience with bricks could not. For that inexperienced builder, the instructions would have to be broken down into far more detail, including how to prepare the foundation, how to mix the mortar, how to ensure that the wall is straight, what to do at the corners, and so on. Here, the original instructions are like a high-level source program; the detailed breakdown of the instructions is like an object program. A computer is much like a completely inexperienced builder in that its instructions must ultimately be broken down to a very low level of detail before the computer can execute them.

The significant point is that once the brick-wall instructions are broken down in this way, the precise wording and some of the sequence of the original instructions about height, width, length and pattern may well be lost. The lengthier the original, high-level instructions, the more certain it is that the detailed instructions cannot be used to reconstruct them. It is the same with computer programs. Why this inability to convert from a very detailed expression back to a high level one makes a difference will be discussed later.⁹

OBJECT PROGRAMS UNDER THE 1976 COPYRIGHT ACT

Source programs are unquestionably protected by copyright; the rub comes with object programs. The first problem with object programs is determining what they are in the terms of the 1976 Copyright Act. Curiously, courts have addressed just about every other problem surrounding object programs but this one. Among the issues that have already been raised and resolved are these:

(1) Whether object programs are readable by human beings and whether readability is relevant to copyrightability? They are readable only by highly trained experts, but the 1976 Act does not require readability because the Act overturned *White-Smith v. Apollo*;¹⁰

8. See *infra* note 61 and accompanying text.

9. See *infra* text accompanying note 75.

10. *White-Smith Music Publishing Co. v. Apollo Co.*, 209 U.S. 1 (1908), had held that a player piano roll was not a "copy" of the musical composition it caused to be played, because a composition "is not susceptible of being copied until it has been put in a form which others can see and read." 209 U.S. at 17. The House Report on the 1976 Copyright Act observed that the "broad language [of the 1976 Act] is intended to avoid the artificial and largely unjustifiable distinctions, derived from cases such as *White-Smith Publishing Co.* . . . , under which statutory copyrightability in certain cases has been made to depend upon the form or medium in which the work is fixed. Under the bill [later enacted] it makes no difference what the form, manner, or medium of fixation may be—whether . . . it is capable of perception directly or by means of any machine or device" H.R. REP. NO. 1476, 94th Cong., 2d Sess. 52, reprinted in 1976 U.S. CODE CONG. & AD. NEWS 5659, 5665. See, e.g., *Apple Computer, Inc. v. Formula Int'l, Inc.*, 725 F.2d 521, 524-25 (9th Cir. 1984); *Apple Computer, Inc. v. Franklin Computer Corp.*, 714 F.2d 1240, 1248 (3d Cir. 1983), cert. dis-

(2) Whether object programs in a computer chip or "read-only-memory," are fixed in a tangible medium of expression? They are, because they can be perceived or communicated from the chip;¹¹ and

(3) Whether object programs on a chip are so much a part of the computer as a machine that they are functional, mechanical devices and hence outside the scope of copyright's subject matter? They are not, because they are no more a part of a machine than is a musical work on a record or tape player.¹²

In addition to decisions resolving various issues in favor of protecting object programs, the 1976 Act's legislative history shows that Congress intended to provide exactly that protection. The major element of that history is the final report of a congressionally-appointed National Commission on New Technological Uses of Copyrighted Works (CONTU).¹³ The CONTU final report concluded that computer programs should be copyrightable; the conclusion made no distinction between source and object programs.¹⁴ Congress ultimately adopted the CONTU recommendation almost verbatim in a 1980 amendment to the 1976 Act, including in the statute a definition for "computer program" that seems to incorporate both source and object programs.¹⁵ Several federal courts have determined, with little difficulty, that CONTU intended, as must have the Congress, that both source and object programs be protected by copyright.¹⁶

Neither CONTU nor the case law, however, has yet determined the proper copyright category for object programs. The CONTU majority spoke of "computer programs" with but a few express references to source or object programs throughout the Final Report.¹⁷ In a dissenting opinion, Commissioner Hersey went to considerable lengths to distinguish these two

missed, 104 S. Ct. 690 (1984); *Midway Mfg. Co. v. Strohon*, 564 F. Supp. 741, 750-51 (N.D. Ill. 1983).

11. See, e.g., *Apple Computer, Inc. v. Franklin Computer Corp.*, 714 F.2d 1240, 1249 (3d Cir. 1983), *cert. dismissed*, 104 S. Ct. 690 (1984); *Williams Elecs., Inc. v. Artic Int'l, Inc.*, 685 F.2d 870, 874 (3d Cir. 1982); *Midway Mfg. Co. v. Artic Int'l, Inc.*, 547 F. Supp. 999, 1007-08 (N.D. Ill. 1982), *aff'd*, 704 F.2d 1009, 1012 (7th Cir. 1983), *cert. denied*, 104 S. Ct. 90 (1983); *Tandy Corp. v. Personal Micro Computers, Inc.*, 524 F. Supp. 171, 173 (N.D. Cal. 1981).

12. See, e.g., *Apple Computer, Inc. v. Franklin Computer Corp.*, 714 F.2d 1240, 1249, 1251 (3d Cir. 1983) (quoting National Commission on New Technological Uses of Copyrighted Works (CONTU), Library of Congress, Final Report 21 (1978)), *cert. dismissed*, 104 S. Ct. 690 (1984); *Midway Mfg. Co. v. Strohon*, 564 F. Supp. 741, 751-52 (N.D. Ill. 1983). But see Comm'r Hersey's dissent to the CONTU Final Report [hereinafter cited as CONTU Report].

13. Act of Dec. 31, 1974, Pub. L. No. 93-573, tit. II, 88 Stat. 1873 (establishing CONTU).

14. CONTU Report, *supra* note 12, at 1, 12, 14.

15. Act of Dec. 12, 1980, Pub. L. No. 96-517, ch. 38, § 10(a), (b), 94 Stat. 3028 (amending 17 U.S.C. §§ 101, 117 (1976)). The amendment adopted this definition: "A 'computer program' is a set of statements or instructions to be used directly or indirectly in a computer in order to bring about a certain result." Though the legislative history to the 1980 amendment is silent on this point, see H.R. REP. NO. 1307, 96th Cong., 2d Sess. 23, *reprinted in* 1980 U.S. CODE CONG. & AD. NEWS 6460, 6482-83, Congress evidently intended "direct or indirect" to refer to object programs and source programs respectively. *Accord*, *Apple Computer, Inc. v. Formula Int'l Inc.*, 725 F.2d 521, 524-25 (9th Cir. 1984); *Midway Mfg. Co. v. Strohon*, 564 F. Supp. 741, 749-50 (N.D. Ill. 1983).

16. E.g., *Apple Computer, Inc. v. Formula Int'l Inc.*, 725 F.2d 521, 525 (9th Cir. 1984); *Apple Computer, Inc. v. Franklin Computer Corp.*, 714 F.2d 1240, 1249 (3d Cir. 1983), *cert. dismissed*, 104 S. Ct. 690 (1984); *Midway Mfg. Co. v. Strohon*, 564 F. Supp. 741, 750 (N.D. Ill. 1983).

17. The CONTU Report mentions "source" and "object" programs in only three places: pages 21 & n.109, and page 25. In none of these discussions is the distinction suggested to be relevant to copyright.

versions of a program for copyright purposes. He noted three possible classifications of object programs: derivative works, literary works, and copies; each of the three proposed at one time or another by the CONTU subcommittee reporting on software.¹⁸ Because the software subcommittee wanted copyright protection to attach to object programs, its shifting terminology must reflect an uncertainty not about the proper statutory treatment, but about how the wording of the 1976 Act should be used to achieve that treatment. The Commission's failure to do more than just mention object programs in the final report,¹⁹ particularly in the face of Commissioner Hersey's explicit discussion in his dissenting opinion, strongly suggests that the problem of statutory classification of object programs proved intractable and was simply ignored.

Why is the classification of object programs into a statutory pigeon hole so difficult, and why does it matter? It matters because some pigeon holes appear to offer no protection, and proponents of copyright protection naturally would like to avoid them. Why the classification is difficult can only be seen if one goes through the possible theories one by one. There are at least six possibilities.

Theory One: Independently Copyrightable Literary Works

Calling object programs "literary works" perhaps makes the most sense in terms of the Act and its legislative history, and is not inconsistent with calling them derivative works as well. A derivative work, after all, may also be a "literary work." Literary works are defined in the 1976 Act as "works . . . expressed in words, numbers, or other verbal or numerical symbols or indicia"²⁰ Certainly both source and object programs meet this broad definition. Indeed, the House Report on the 1976 Act, quoted in CONTU's final report, says expressly that "the term 'literary works' includes . . . computer programs to the extent that they incorporate authorship in the programmer's expression of original ideas, as distinguished from the ideas themselves."²¹

Here again is the absence of any distinction between source programs and object programs. The point to focus on, of course, is the House Report's emphasis on the need for a "literary work" to have "authorship in the programmer's expression of . . . ideas." That requirement might limit "literary works" to any work that incorporates "the programmer's expression," and hence to the only work created by the *programmer*, the source program. Under this reading of the House Report, the category "literary work" would not include the work created by the *computer*, the object program. A reading of the Report in context, however, shows that the distinction sought was not that between "programmer's expression" and "computer's expression," but was rather the familiar copyright distinction between the expression of

18. CONTU Report, *supra* note 12, at 32.

19. See *supra* note 17.

20. 17 U.S.C. § 101 (1982).

21. H.R. REP. NO. 1476, 94th Cong., 2d Sess. 54, reprinted in 1976 U.S. CODE CONG. & AD. NEWS 5659, 5667, quoted in CONTU Report, *supra* note 12, at 16.

an idea and the idea itself.²² The language of the House Report therefore ties in to the proscriptions in section 102(b) of the 1976 Copyright Act against copyright protection of "any idea, procedure, process," etc. and does not qualify section 101's definition of "literary works."²³

In sum, classifying object programs as literary works makes perfect sense. That classification suggests the theory that object programs are copyrightable as independent, original works of authorship. But surely this theory fails: no author, at least no human author, has created anything in an object program. All the originality, the authorship, the expression, and the creativity are vested in the source program. The object program is mechanically produced without an author's contribution at all. No court has even suggested that an object program reflects creativity or expression independently from its source program.

Calling object programs literary works therefore makes sense only in the abstract; it does not provide a theory for the protection of object programs.

Theory Two: Independently Copyrightable Derivative Works

Several courts have implied, however, that object programs are derivative works and can achieve copyright protection under that label.²⁴ A derivative work is one that is "based upon one or more preexisting works, such as a translation"²⁵ The computer's translation of the source program creates a new work that is "based upon" a pre-existing work; hence the object program may be considered a derivative work. Indeed, it would be difficult to classify an object program as anything else. The House Report notes that "the terms 'compilations' and 'derivative works' . . . comprehend every . . . work that employs pre-existing material or data of any kind."²⁶ An object program not only "employs" preexisting data; it is made entirely from pre-existing data.

22. The House Report goes on to discuss the expression-idea distinction specifically discussing the concern that copyright in programs might unjustifiably "extend . . . to the methodology or processes" of the program. There is, however, no further discussion of any "programmer expression" versus "computer expression" distinction. H.R. REP. NO. 1476, 94th Cong., 2d Sess. 57, reprinted in 1976 U.S. CODE CONG. & AD. NEWS 5659, 5670.

23. See 17 U.S.C. §§ 102(b), 101 (1982).

24. E.g., *Apple Computer, Inc. v. Franklin Computer Corp.*, 714 F.2d 1240, 1248, 1251-52 (3d Cir. 1983) (calling object programs a translation and implying their derivative status); *cert. dismissed*, 104 S. Ct. 690 (1984); *Midway Mfg. Co. v. Strohon*, 564 F. Supp. 741, 750-51 (N.D. Ill. 1983) ("object code is nothing other than a direct transformation of a computer program composed . . . in source code" (emphasis added)); *Tandy Corp. v. Personal Micro Computers, Inc.*, 524 F. Supp. 171, 173 (N.D. Cal. 1981) (calling object programs a "translation"). See also *Apple Computer Inc. v. Computer Edge Pty. Ltd.*, 28 PAT., TRADEMARK & CORP. J. 256 (B.N.A. July 12, 1984) (Federal Court of Australia concludes that object programs can be protected as "translations" or "adaptations" of source code); Argy, *Legal Protection of Computer Software in Australia?* 1 J. LAW & INFORMATION SCIENCE 256, 259 (1983) (concluding that object programs are translations within the meaning of the Australian copyright act) (the JOURNAL OF LAW & INFORMATION SCIENCE is published at the New South Wales Institute of Technology in Australia).

25. 17 U.S.C. § 101 (1982).

26. H.R. REP. NO. 1476, 94th Cong., 2d Sess. 57, reprinted in 1976 U.S. CODE CONG. & AD. NEWS 5659, 5670. The House Report actually says that the terms "compilations" and "derivative works" include "every copyrightable work that employs pre-existing material. . . ." (emphasis added). If an object program is not itself copyrightable, as this paper argues, then this phrase suggests that an object program cannot be a derivative work. But the word "copyrightable" either means

Derivative works are part of the subject matter of copyright and may sustain a copyright independent of the original, or underlying, work.²⁷ An object program is therefore, by virtue of its status as a derivative work, eligible for copyright protection, but only if it meets the usual copyright requisites of being an "original work[] of authorship fixed in any tangible medium of expression, . . . from which [the work] can be perceived, reproduced, or otherwise communicated, either directly or with the aid of a machine or device."²⁸ In short, if both the underlying source program and the derivative object program are to sustain separate copyrights, then each must independently meet the test of being an original work of authorship.²⁹ The House Report notes the well-established doctrine that copyright on a derivative work "covers only the material added by the *later author*"³⁰ Whatever material was present originally in the underlying work is protected in the derivative work, if protected at all, by the underlying work's copyright, not by a separate derivative work copyright.

Once again, even as a derivative work, an object program fails the test of being an original work of authorship. It has no "later author," other than the underlying source program's author; it cannot be an independent work of authorship because it is produced automatically by a machine. It is therefore not copyrightable by the plain terms of the statute.³¹ The legislative history also suggests, though in the context of sound recordings, that a work created by "purely mechanical means" lacks originality and cannot achieve copyright protection for that reason.³² An object program is, of course, cre-

something else in this context, or its use was inadvertent: elsewhere the House Report shows that derivative works need not be copyrightable. See *infra* text accompanying notes 47-48.

In some circumstances, programs are pieced together from snippets of other programs, and thus might be called compilations instead of derivative works. But this piecing together is almost always done in the source code version; it provides no new theory for the protection of object programs.

27. 17 U.S.C. § 103(a) (1982).

28. 17 U.S.C. § 102(a) (1982). See also H.R. REP. NO. 1476, 94th Cong., 2d Sess. 51-57, reprinted in 1976 U.S. CODE CONG. & AD. NEWS 5659, 5664-70.

29. See 17 U.S.C. § 103(b); H.R. REP. NO. 1476, 94th Cong., 2d Sess. 57, reprinted in 1976 U.S. CODE CONG. & AD. NEWS 5659, 5670.

30. H.R. REP. NO. 1476, 94th Cong., 2d Sess. 57, reprinted in 1976 U.S. CODE CONG. & AD. NEWS 5659, 5670 (emphasis added). See also *Russell v. Price*, 612 F.2d 1123, 1128 (9th Cir. 1979), cert. denied, 446 U.S. 952 (1980).

31. *Accord*, L. Batlin & Son, Inc. v. Snyder, 536 F.2d 486, 490-91 (2d Cir. 1976), cert. denied, 429 U.S. 857 (1976), discussed in this same vein by Stern, *supra* note 3, at 14 n.53. Stern concludes that object programs cannot be derivative works in part because the object programs are not fixed. Stern, *supra* note 3, at 14. This conclusion is wrong for two reasons: first, object programs can indeed be fixed, see *supra* text accompanying note 11; second, the 1976 Act does not impose a requirement of fixation on derivative works. See *infra* text accompanying notes 47-48.

32. H.R. REP. NO. 1476, 94th Cong., 2d Sess. 56, reprinted in 1976 U.S. CODE CONG. & AD. NEWS 5659, 5669. Whether this phrase ought to apply to object programs is debatable. Doubtless Congress intended that a mechanical fixation would involve no changes and would be a duplication of some preexisting work; hence, the fixation would lack originality. An object program, though mechanically fixed, can still be one of a kind and perhaps could therefore be termed "original." Yet, the term "original" is often used by courts to refer to a requirement for creative authorship and not just to "an absence of copying." In this sense, then, the House Report may be saying exactly what the discussion above says: purely mechanical fixation lacks authorship.

One court has, for example, used the term "originality" in just this way to deny copyright to a translation of a list of English words into Arabic. See *Signo Trading Int'l Ltd. v. Gordon*, 535 F. Supp. 362 (N.D. Cal. 1981). Though recognizing the possibility that the initial list of English words might be copyrightable, *id.* at 364, the court concluded that "the translation of the word list [into Arabic] . . . is a fairly mechanical process, requiring little if any originality," *id.*, and so was not

ated by "purely mechanical means." Object programs therefore fail the requirements for independent protection as derivative works.

The CONTU's Final Report indirectly supports this conclusion in a discussion of the use of a computer to create new works, such as stories, poems, drawings, music, etc. CONTU noted that a musical work created by a computer could indeed be copyrightable, as long as the would-be copyright owner "exercised sufficient control over the production of the work to be considered its author."³³ However much control is sufficient for authorship under this test, it is plain that some control is required—and a source program author typically has no control over the translating program's production of object programs.³⁴

Speaking more generally, CONTU noted that the copyrightability of "a work created through application of computer technology . . . depends not on the device or devices used in its creation, but rather upon the presence of at least minimal human creative effort at the time the work is produced."³⁵ Though not addressing the copyright status of object programs as such, CONTU's remarks are entirely consistent with copyright doctrine generally and should certainly apply to the use of a computer to produce object programs. Just as certainly, the production of an object program is not accompanied by "minimal human creative effort at the time."

In short, object programs fail to meet the requirements for copyrightability either as independent literary works or as derivative works.

Theory Three: Joint Works

Yet another theory is that object programs should be considered "joint works" under the 1976 Act.³⁶ The translation process on a computer involves three programs: the source program, the translating program, and the resulting object program. Typically, the source program and the translating program are written by different people. The joint works theory suggests that the creations of the source program author *and* the author of the translating program combine to form a new work, the object program. Perhaps if both authors are in the picture, the object program's lack of independent creativity or originality will cease to be a problem.

The author of the translating program has a lot to do with the object programs produced by that program. The source program's author has nothing to do with the translating process, but he does provide the "raw

copyrightable. The plaintiff in *Signo* apparently used human translators to make the translation; if human translation can be a mechanical process, surely a computer translation would be even more so.

33. CONTU Report, *supra* note 12, at 46 n.188.

34. Programmers may be offered some options, such as faster translation in exchange for a slower-running object program, or vice-versa, or to obtain certain printouts of the program as it is translated, or to obtain printouts of the object program version. Other than the first, rarely offered option, these opportunities for variation are trivial and have nothing to do with control over the object program itself. Even with the first option, "control" takes the form of a desire for a result, not the exercise of means or expressions to reach that result.

35. CONTU Report, *supra* note 12, at 45.

36. This theory was suggested by the late Professor Alan Latman, of New York University, at a conference titled "Software Protection" held in Washington, D.C., March 9, 1984.

material" for that process; it makes sense to say that both parties substantially contribute to the finished product.

Case law interpretations of joint works show that joint authors need not work together or even at the same time. One author can write the lyrics for a song, then later a composer unknown to the lyricist can add the music. The result is a joint work.³⁷ The "joint work" cases show, in sum, that one author can complete a work, can turn the product of his effort over to someone else, and with no further effort or authorship on his part can become one of the owners of copyright in a new, joint work. Object programs seem to fit this pattern quite well: the source program author completes the source program, turns it over to someone (or something) else, and with no further effort, authorship, or creativity on his part, a new, joint work is created. The objection to copyrighting object programs, that they lack authorship, is indeed mitigated by this view: the extra ounce of necessary authorship comes from the author of the translating program.

The primary test of joint authorship is that the authors intend "that their contributions be merged into inseparable or interdependent parts of a unitary whole."³⁸ Source program authors certainly intend their programs to be translated and run on a computer; just as certainly, authors of translating programs intend that those programs be used to translate the source programs of others. Even the intent test of joint authorship seems to be met by object programs.

Two drawbacks to the joint work theory, though, are compelling. First, can Congress really have intended that one person, the author of the translating program, through a one-time effort in writing that program, should thereafter become a joint author of thousands of object programs produced by thousands of unknown programmers using his translator? Regardless of Congress's intent, does it make sense to give the translating program's author a right to a share in the profits of every single program translated by his program and then sold? If this right were not waived, the practical difficulties of negotiating agreements with thousands or tens of thousands of source program authors would be enormous. More likely, sellers of translating programs would simply waive their rights of joint authorship. Few programmers would expect, in the first place, that the author of a translating program would be a co-owner of any object programs they produced. To discover simultaneously that such a right existed but had been waived would be confusing indeed, even silly. If this kind of automatic waiver did evolve as the rule, the possibility of joint authorship would have provided no additional incentive to the creation of translating programs—leaving the situation just as it is right now. And if reliance on the joint authorship theory of computer program copyrights would provide no additional incentives to anybody, but would simply require the printing of boilerplate waivers, then the theory obviously leaves much to be desired.

From another perspective, the joint authorship theory simply fails out-

37. See, e.g., *Edward B. Marks Music Corp. v. Jerry Vogel Music Co.*, 140 F.2d 266, 267 (2d Cir. 1944).

38. 17 U.S.C. § 101 (1982).

right. The test of joint authorship is the authors' intent that their separate contributions be "merged."³⁹ A merger can be into an inseparable whole, like a movie on which script writers, actors, producers, and a host of others collaborate;⁴⁰ or it can be into "interdependent parts" of a whole, like a song on which both a composer and a lyricist have joined efforts.⁴¹

Whatever the type of merger, both the case law and the Copyright Act unarguably say that some kind of merger must happen. That merger has, in every case addressing the issue, been a merger of expressions of authorship.⁴² Whatever it is that has "merged" into an object program, it does not look like two expressions. The contribution of the translating program's author was to express the rules for translation in the form of a computer program. The object programs produced by that program reflect not the expression of those rules, but the end result of the operation of the rules: a translation. The operation of rules is not the "expression" that copyright is supposed to protect. Courts considering protection of the rules for games, for example, uniformly allow protection only for the written description of the rules, not for the rules in the abstract.⁴³ As CONTU notes, owners of copyright in the written rules of a game cannot stop others from using the rules to play the game;⁴⁴ nor can the copyright owner of a description of a system of accounting own a copyright in the execution of the system.⁴⁵ Because the operation of rules about translating is not a copyrightable expression, there can be no merger of expressions in the object program, and that program cannot be a joint work.

Once again, CONTU's Final Report, in discussing copyright protection for computer-created works, supports this conclusion. CONTU briefly examined the situation in which one person—a composer, for example—selected the input data to go into a music-composing program written by a second person. Would the second person be a joint author of the resulting musical work? CONTU apparently thought not: "It appears to the Commission that authorship of the [music-composing] program . . . is entirely separate from authorship of the final work. . . ."⁴⁶

The analogy to the production of object programs is striking. The source program author selects his expressions and passes them as input to a translating program. The author of the translating program has created something that is entirely separate from what the source program author has

39. 17 U.S.C. § 101 (1982); 1 M. NIMMER, NIMMER ON COPYRIGHT, Release No. 15, § 6.03, at 6-7 (1983).

40. 1 M. NIMMER, *supra* note 39, § 6.02 at 6-4.

41. *Id.*

42. See, e.g., *Richmond v. Weiner*, 353 F.2d 41, 46-47 (9th Cir. 1965) (involving the co-authorship of a book); *cert. denied*, 384 U.S. 928 (1966); *Shapiro, Bernstein & Co. v. Jerry Vogel Music Co.*, 161 F.2d 406, 409-10 (2d Cir. 1946) (involving the joint composition of a song), *cert. denied*, 331 U.S. 820 (1946); *Mister B Textiles Inc. v. Woodcrest Fabrics, Inc.*, 523 F. Supp. 21, 24-25 (S.D.N.Y. 1981) (involving the joint production of a fabric design). Cf. *Kenbrooke Fabrics, Inc. v. Material Things*, 223 U.S.P.Q. 1039 (BNA) (S.D.N.Y. 1984) (one party's offering of a fabric design suggestion is not a "tangible contribution" to the other party's actual designs; no joint work is created).

43. See, e.g., *Affiliated Enterprises v. Gruber*, 86 F.2d 958, 960-61 (1st Cir. 1936); *Whist Club v. Foster*, 42 F.2d 782 (S.D.N.Y. 1929); *Seltzer v. Sunbrock*, 22 F. Supp. 621, 630 (S.D. Cal. 1938).

44. CONTU Report, *supra* note 12, at 20.

45. *Baker v. Selden*, 101 U.S. 99 (1879).

46. CONTU Report, *supra* note 12, at 45.

"created" by use of the translating program. Hence, the two authors are not joint authors of the resulting object program.

The joint works theory, then, like the independent literary works theory and the independently copyrightable derivative works theory, fails to offer protection to object programs.

OTHER THEORIES OF PROTECTION

The conclusion that object programs are not independently copyrightable does not end the matter, for the Copyright Act may still provide a mechanism for protection. Three other theories of protection can be examined: first, copying an object program is the equivalent of preparing a derivative work; second, an object program is a "copy" of its source program; and third, copying an object program constitutes copying certain protected portions—the "pattern"—of the underlying source program.

Theory Four: The Right to "Prepare" Derivative Works

Authors of original underlying works, such as source programs, have the exclusive right to control the preparation of derivative works, such as object programs, whether or not the derivative work is copyrightable. Both the definition of "derivative work" and the legislative history makes that conclusion plain: a derivative work is "a work based upon one or more pre-existing works. . . ."⁴⁷ Nothing in the definition requires that a derivative work be fixed in a tangible medium, for example, and the House Report expressly acknowledges that an unrecorded improvised dance or skit based on a copyrighted work would constitute a derivative work, even though not independently copyrightable.⁴⁸ Source program authors, then, are able to

47. 17 U.S.C. § 101 (1982).

48. "[P]reparation of a derivative work, such as a ballet, pantomime, or improvised performance, may be an infringement [of § 106(2)] even though nothing is ever fixed in tangible form." H.R. REP. NO. 1476, 94th Cong., 2d Sess. 62, reprinted in 1976 U.S. CODE CONG. & AD. NEWS 5659, 5675. The idea that a work can infringe, even though it is not itself copyrightable subject matter, is not new. Professor Chafee pointed out the example of oral production of plays some forty years ago. See Z. Chafee, *Reflections on the Law of Copyright: I*, 45 COLUM. L. REV. 503, 505 (1945). See also discussion of *Kalem Company v. Harper Brothers*, *infra* text accompanying notes 68-71.

A distinction might, of course, be made between the House Report's example of something not fixed in a tangible medium and hence outside the subject matter of copyright, and something tangible like a copy of an object program that is within copyright subject matter but, for lack of authorship, falls short of the standard of protection. The distinction might be used to justify a conclusion that the latter works should not be controllable by the author as derivative works, even though the former are. This reasoning follows the highly developed patent law distinction between the subject matter protected and the standards for protection of that subject matter, see 1 P. ROSENBERG, *PATENT LAW FUNDAMENTALS*, ch. 6 at 6-2 (1984), but it makes little sense in copyright law.

The Copyright Act itself does not clearly define the subject matter of copyright at all. Section 102, titled "Subject matter of copyright: In general" does no more than say that "protection subsists" in "original works of authorship" that are tangibly fixed, and then gives examples of "works of authorship." By itself, that section suggests, if anything, that the subject matter of copyright is "works of authorship" and that originality and tangibility are not subject matter requirements but standards to be met by proper subject matter. Under that view, it would not matter that the House Report's example of an intangible derivative work was different from the example of an object program that is tangible but lacks authorship: both would show a failure of copyrightability because of a failure to meet the tests of protection.

The question of subject matter versus tests for protection is complicated by the Act's use of the term "subject matter" to refer to several attributes of a work that by any common sense approach

control the preparation of derivative works without regard to whether those works are themselves copyrightable.

If the unauthorized copying of an object program could be said to constitute the "preparation" of a derivative work, program producers would have a means of copyright protection. But can it? A typical example of a derivative work's preparation is that of a person who reads someone else's novel and then, without authorization, turns it into a movie or play. That person has plainly "prepared a derivative work" and can be stopped by the novel's author. Where the author himself or his licensee has prepared the derivative work, say, a movie, and the other person merely copies the movie without ever seeing or caring to see the underlying novel, the situation is also readily resolvable. The author will have a copyright in both the novel and the movie, and can proceed under his right to control the reproduction of the independently copyrighted movie without having to fall back on a theory that his right to prepare a derivative work has been infringed.

The problem arises when the derivative work cannot be protected by copyright. For example, an author writes and hence copyrights a short story. With his approval, a dance troupe improvises an unrecorded dance that is based on the story. Without his approval, a second dance troupe sees the improvised dance and later copies it into one of its own performances. Can the author or the first dance troupe stop the second troupe's performance by arguing that the second troupe "prepared" a derivative work without authority?

A line of cases decided under the 1909 Copyright Act addresses a similar issue with regard to recordings of musical works. Under the Act, a musical composition could be copyrighted, but not a recording of a performance of that composition.⁴⁹ A recording was, however, what would today be called a "derivative work."⁵⁰ The owner of a copyright in a musical composition could control the first recording of a performance of the composi-

would be considered tests for or exceptions to protection. Section 105, for example, explains that if the United States Government authors a literary, dramatic, or any other work, it cannot obtain federal copyright protection for it. This section, too, though it seems to describe exceptions to protection for certain things otherwise well within the subject matter of copyright, is titled "Subject matter of copyright: United States Government works."

The problem of defining the subject matter of copyright might well remain an academic one, but for the reference to it in a later section specifying that certain state laws are preempted by the copyright laws. Section 301(a) says that state laws are preempted if they provide protection "equivalent to any of the exclusive rights within the general scope of copyright . . . in works of authorship that are fixed in a tangible medium of expression *and* come within the subject matter of copyright as specified in sections 102 and 103." (Emphasis added.)

Because United States Government works seem to be excluded from the subject matter of copyright in the title to section 105, but not mentioned in sections 102 or 103, the statute allows the Government to obtain state or common law copyright protection in anything it authors. By the terms of section 301, this state law protection will not be pre-empted. No doubt Congress did not intend this result, but it follows from the Act's choice of section titles.

At any rate, the statute only vaguely indicates what the subject matter of copyright is. Any argument that object programs and impromptu dances are distinguished by whether they are within or without copyright's subject matter thus flies in the face of Congress's indifference to that distinction as a significant one for copyright.

49. 17 U.S.C. § 1 *et seq.* (1909). 1 M. NIMMER, *supra* note 41, § 2.10[A] & [A][1]. As of February 15, 1972, recordings become subject to copyright by amendment of the 1909 Act, *id.*, but the pre-1972 situation is of interest here.

50. 1 M. NIMMER, *supra* note 41, § 3.01 at 3-2, § 8.04[E] n.45 at 8-62.

tion.⁵¹ Once that authorized recording was made, however, anyone could obtain a compulsory license from the owner to "make similar use of the copyrighted work upon the payment . . . of a royalty" to that owner.⁵²

Thus, the 1909 Act plainly allowed a compulsory license for anyone who wanted to hire an orchestra and make a recording; but eventually the question arose whether anyone could skip the orchestra hiring altogether and simply duplicate an existing record. In a 1972 case, *Duchess Music Corp. v. Stern*,⁵³ the Ninth Circuit concluded that simple duplication was unlawful. The facts neatly parallel the case of computer programs: a record, not itself copyrightable, could nevertheless not be copied by others. With computer programs, an object program is not itself copyrightable, yet the analogy argues for the copyright owner's authority to control the reproduction of object programs.

The analogy is attractive and useful, though it is not compelling. First, the 1909 Act expressly granted copyright owners the right to control the making of a record of their musical compositions. Further, the Act set up the compulsory licensing scheme. Congress carried over in modified form that same scheme for musical compositions and records into the 1976 Act,⁵⁴ but did not provide a similar scheme for computer programs. That Congress, through the CONTU report, was well aware of computer programs by 1976 and did not make the sort of arrangement for programs that it made for musical compositions and records⁵⁵ at least argues against a court's so doing.

Second, a reasonable reading of the 1909 Act shows that the *Duchess Music* decision was simply incorrect. The 1909 Act had said that allowing copyright holders in musical compositions the right to control the first use of their compositions for recording was in exchange for allowing others a compulsory license. "[W]henver the owner of a musical copyright has [allowed the composition to be recorded], any other person may make similar use of the copyrighted work" on compliance with requirements for payment of a royalty.⁵⁶ The Ninth Circuit concluded, surprisingly, that making "similar use" of a work did not include making "exact and identical copies" of it.⁵⁷

In short, *Duchess Music* construed the compulsory license provisions to apply only to making a new recording, not to duplicating an existing recording. Though perhaps a desirable and sensible result, the decision did not square with contemporary understanding of the licensing provisions.⁵⁸

51. Section 1 of the 1909 Act allowed the owner of copyright in a musical work the exclusive right "to make any arrangement or setting of it or of the melody of it in . . . any form of record . . . from which it may be read or reproduced." 17 U.S.C. § 1 (1909).

52. *Id.*

53. *Duchess Music Corp. v. Stern*, 458 F.2d 1305 (9th Cir. 1972).

54. 17 U.S.C. § 115 (1976).

55. The 1976 Act refers to musical compositions as "musical works," and records (and tapes and other devices) as "sound recordings," 17 U.S.C. § 102 (1976).

56. 1909 Copyright Act, 17 U.S.C. § 1 (1909).

57. *Duchess Music Corp.*, 458 F.2d at 1310.

58. Dissenting, Judge Byrne noted, for example, that Congress had enacted amendments to the Copyright Act in 1972 precisely because without the amendments "record pirates . . . 'can and do engage in widespread unauthorized reproduction of phonograph records and tapes without violating Federal copyright law.'" *Duchess Music Corp.*, 458 F.2d at 1311 (Byrne, J., dissenting and quoting

Other courts did follow the case,⁵⁹ however, so it remains good law, though unnecessary law in view of Congress's 1971 amendments allowing greater protection for recordings.⁶⁰

In any event, *Duchess Music* provides some basis for saying that the owner of copyright in a source program should be able to prevent anyone from copying the object program that derives from it. The principle of the case has in fact been extended outside the bounds of music to literary and dramatic works under the 1976 Act. Just last year, the Ninth Circuit reaffirmed *Duchess Music* in *Lone Ranger Television, Inc. v. Program Radio Corp.*⁶¹ and, in doing so, drew analogies to the protection of an uncopyrighted film that derived from a copyrighted novel.⁶²

Lone Ranger TV itself offers additional support for the theory that copying an object program is the equivalent of preparing a derivative work. The plaintiff in that case owned the copyrights to various Lone Ranger radio scripts but conceded the absence of copyright protection in audio tapes that recorded performances of the scripts. The defendant had taken some of those tapes and "re-mixed the recordings into broadcast cartridges for radio play."⁶³ Over the defendant's arguments that copying uncopyrightable tapes could not violate copyright law, the court drew on *Duchess Music* to conclude, as an interpretation of the 1976 Act, that the copying constituted the preparation of a derivative work.⁶⁴

The facts of *Lone Ranger TV* make a slightly better case for finding an unlawful preparation of a derivative work than the computer program facts do. The defendant had, after all, "re-mixed" the recordings and hence done more than just copy them. Yet the court put no stress on the fact of mixing, and relied on *Duchess Music*, which did involve simple copying, without making the distinction.

This line of cases, then, involving recordings under both the 1909 and 1976 Acts, strongly supports the theory that copying an uncopyrightable object program infringes the source program owner's right to prepare derivative works. A similar line of cases provides inconsistent support. This second line considers the copying not of uncopyrightable derivative works, but of copyrightable derivative works that have lost copyright protection. One such case is the Second Circuit's 1977 decision in *Rohauer v. Killiam Shows*,

House Report 92-487) (emphasis added by Judge Byrne). Nimmer also roundly criticizes the decision. See 1 M. NIMMER, *supra* note 39 § 8.04[E] at 8-60 through 8-64 & 8-60 at 39.

59. See, e.g., *Fame Publishing Co. v. Alabama Custom Tape, Inc.*, 507 F.2d 667 (5th Cir. 1975); *Jondora Music Publishing Co. v. Melody Recordings, Inc.*, 506 F.2d 392 (3d Cir. 1974); *Edward B. Marks Music Corp. v. Colorado Magnetics, Inc.*, 497 F.2d 285 (10th Cir. 1974), all cited in 1 M. NIMMER, *supra* note 39, § 8.04[E] at 8-60 n.40.

60. Pub. Law No. 92-140, Oct. 15, 1971, effective as of February 15, 1972, defined away the composition-record dilemma by defining records as copies of the compositions they embody. See 17 U.S.C. § 101(e) (1972).

61. 740 F.2d 718 (9th Cir. 1984).

62. *Id.* at 722-23. The film case, *Russell v. Price*, 612 F.2d 1123 (9th Cir. 1979), *cert. denied*, 446 U.S. 952 (1980), is almost identical to *Filmvideo Releasing Corp. v. Hastings*, 509 F. Supp. 60 (S.D.N.Y. 1981), *aff'd*, 668 F.2d 91 (2d Cir. 1981), discussed under the "pattern" theory of protection *infra* at notes 93-97.

63. *Lone Ranger Television*, 740 F.2d at 720.

64. *Id.* at 722.

Inc.,⁶⁵ decided under the 1909 Copyright Act.

Rohauer dealt with a fairly complicated factual setting. In brief, the owner of an independently copyrighted movie, *Son of the Sheik*, derived from a copyrighted 1925 novel under a license from the novel's author, made and showed a videotape copy of the movie. The movie owner's license to make any new movies from the novel had lapsed. The owner of the underlying novel's copyright argued that copying the movie onto videotape and exhibiting the tape infringed the owner's rights in the novel. The court's opinion dealt primarily with the question whether exhibiting the tape "dramatized" the novel and hence deprived the novel owner of one of the incidents of copyright ownership.

The court noted in addition that copying the film onto videotape, even with minor changes in titles and with the addition of new music, did not constitute the creation of a "new version" of the movie.⁶⁶ Therefore, the court implicitly found that the making of the videotape was merely a copying of the movie, and hence did not infringe the novel owner's rights to make further derivative works based on the novel. *Rohauer* suggests a conclusion that making a copy of a derivative work, such as an object program, does not equal making a new derivative work, and hence cannot be controlled under the source program creator's right to control the "preparation of derivative works."

Rohauer does not, however, compel that conclusion primarily because the issue was not in the least squarely presented, and secondarily because the case was based partly on the need to protect the movie creator's substantial original effort.⁶⁷ The very crux of the problem of protecting object programs is that they require no original effort. "Copying" a work and "preparing a derivative" work are quite distinct when applied to copying a movie and preparing a movie from a novel. The distinction almost disappears, however, when applied to object programs. To the person sitting in front of a computer, there is no practical difference at all between preparing an object program and copying that program, other than the time it takes to complete either operation.

A better case to support the conclusion that copying a derivative work equals preparing a derivative work is the 1911 Supreme Court case, *Kalem Company v. Harper Brothers*.⁶⁸ Under the 1891 Copyright Act applicable to the case,⁶⁹ authors had the right to control dramatizations of their works.⁷⁰ A dramatization, at that time, evidently consisted of live actors reciting dialog. The Kalem Company was in the business of making motion pictures. Without permission, it prepared its own screenplay of the plaintiff's novel, *Ben Hur*, filmed it, and exhibited it widely. *Ben Hur*'s author sued Kalem on the theory that every exhibition of the film was a dramatization of the

65. 551 F.2d 484 (2d Cir. 1977), *cert. denied*, 431 U.S. 949 (1977).

66. *Rohauer*, 551 F.2d at 494 n.12. *But see* 1 M. NIMMER, *supra* note 39, § 3.03 at 3-12 n.19 (criticizing *Rohauer*'s conclusion on this point).

67. *Rohauer*, 551 F.2d at 493.

68. 222 U.S. 55 (1911).

69. Act of March 3, 1891, c. 565, 26 Stat. 1106.

70. *Id.*

novel. *Kalem* defended on the grounds that they had simply made pictures of various staged scenes from the novel, which did not constitute a "dramatization."

The Supreme Court held that the movie exhibitions did constitute dramatizations of the underlying novel.⁷¹ Thus, *Kalem* offers support for the theory that copying an object program is the equivalent of preparing a derivative work. Under the 1891 Copyright Act, as under the 1976 Act, an author could not copyright a live performance as such. Similarly, an author cannot copyright an object program as such. Under the 1891 Act, an author could, however, control the preparation of the live performance as a dramatization of an existing copyrighted work. With computer source programs, an author can control the preparation of an object program as a derivative work. Finally, *Kalem* gave authors the right to control the copying of a dramatization onto a permanent medium, film, because the showing of the film constituted the doing of the controlled act: the dramatization. By parallel reasoning, source program authors should have the right to control the copying of a derivative work, the object program, because the making of the copy constitutes the doing of the controlled act: the preparation of the object program as a derivative work.

Theory number four thus proves useful for protecting object programs. The unauthorized copying of an object program infringes the source program author's right to control the preparation of derivative works based on that source program.

Theory Five: Copies of Source Programs

If copying an object program and preparing an object program from a source program should be treated indistinguishably, as theory four suggests, there is still the choice of treating both operations as a copying, rather than treating both as a preparing. If both operations can be called "copying," then object programs are "copies" of their source programs. If an object program is a copy of a copyrightable source program, the copyright law ensures that no one may copy that copy. Protection for object programs is then assured.

At least one federal district court has implied that object programs are just copies. The opinion in the Northern District of California case, *GCA Corp. v. Chance*,⁷² says that both source and object versions "are to be treated as one work."⁷³ If only one work is in issue, then the two versions, in copyright terminology, must be "copies" of it.

Unfortunately, the definitions in the 1976 Act make it hard to describe an object program as a "copy" of its source program. A "copy," under the act, is a "material object[] . . . in which a work is fixed . . . and from which the work can be perceived. . . ."⁷⁴ The primary obstacle to calling even a particular, tangible fixation of an object program a "copy" is that

71. *Kalem*, 222 U.S. at 61.

72. 217 U.S.P.Q. 718 (N.D. Cal. 1982).

73. *Id.* at 720.

74. 17 U.S.C. § 101 (1982).

"the work" cannot, in most instances, be perceived from it. "The work," refers presumably, to the work created by the programmer. A program written in a high-level language, like the brick-wall instructions example, once translated into the details of an object program, cannot be perceived from that object program.

To be sure, an object program can be run through a "de-compiler" to produce a version of the program that is more easily readable by programmers than the ones and zeroes of object code. But the "de-compiled" version in no sense resembles the expression originally authored by a programmer working in a language like Basic or Cobol. Indeed, absent some special identifying information supplied by the translator program, one cannot examine an object program (or its de-compiled version) and tell what language the original source program was written in, let alone the particular expression chosen from that language by the programmer.

In the margin are examples of two different source programs written in the Basic programming language.⁷⁵ The two examples look different, though they both yield exactly the same object program. An expert knowing that a source program was written in Basic could tentatively reconstruct that program from the object program shown, but obviously could not know which of the two versions, or which of other possible variations, the original programmer had selected. When so little of the programmer's expression remains in the object code, it makes little sense to say that that expression—the "work" authored by the programmer—can be perceived from the object code. If the work cannot be perceived, then the disc or tape or silicon chip containing an object program cannot be a "copy" of the source program within the definitions of the Copyright Act.

Theory five, that object programs should be treated as "copies" of their source programs, is therefore unsuccessful.

Theory Six: Reflections of the Source Program's "Pattern"

The sixth approach to protecting object programs relies on cases like

75. Program 1

```

1  x = 1
2  if not x <= 9 then 6
3  print x
4  x = x + 1
5  go to 2
6  end

```

Program 2

```

100  rem — this program prints the
200  rem — numbers from one to nine
300  '
400  let number = 1
500  while number <= 9
600    print number
700    number = number + 1
800  wend
900  end

```

Each of these programs, when translated to object form on an IBM personal computer, yields the same object program (shown here as "de-compiled" with the program called "Debug"):

*Rohauer*⁷⁶ that have specifically addressed the relationship of derivative works to their underlying works. The cases suggest a theory for finding that making copies of or performing a derivative work can infringe rights in the underlying work. The theory relies on the notion that an author's copyright protects more than just the literal wording of, for example, a novel, but extends also to the story line, plot, or "pattern" of the work.

Rohauer is only one of several cases that has wrestled with this theory, and one of the few that has rejected it.⁷⁷ All the notable cases were decided under the 1909 Act,⁷⁸ but the 1976 Act does not much change the boundaries of the theory.⁷⁹ It will be useful, therefore, to review these earlier cases.

001A	CALL	0925:0000
001F	PUSH	BP
0020	MOV	BP,SP
0022	SUB	SP,0000
0026	CALL	091C:004D
002B	MOV	DI,0C42
002E	MOV	SI,0C50
0031	INT	A6
0033	MOV	DI,0C54
0036	MOV	SI,0C42
0039	INT	CA
003B	JBE	0040
003D	JMP	0056
0040	INT	E6
0042	MOV	BX,0C42
0045	INT	96
0047	MOV	DI,0C50
004A	MOV	SI,0C42
004D	INT	AA
004F	MOV	DI,SI
0051	INT	A8
0053	JMP	0033
0056	INT	3E

76. See discussion *supra* text accompanying notes 65-67.

77. In one other case, the court rejected an argument that the pattern of an underlying work was copied into a derivative work and was infringed by a showing of the derivative work. See *Classic Film Museum, Inc. v. Warner Bros., Inc.*, 453 F. Supp. 852 (D. Me. 1978), *aff'd*, 597 F.2d 13 (1st Cir. 1979). In *Classic Film*, Warner Brothers owned a common law copyright in the unpublished story and screenplay for "A Star is Born." A 1937 movie made from the screenplay had been copyrighted, but protection lapsed without renewal in 1965. Warner Brothers argued that any unauthorized showing of the movie infringed its potentially perpetual common law rights in the story and screenplay. The district court concluded that the perpetual protection of the movie that would result from this argument conflicted with the Constitutional requirement that copyrights subsist for a limited time. 453 F. Supp. at 855. The case may have been correctly decided as a matter of pre-1976 Copyright Act law, but by the time the district court's decision was issued, the basis for the decision was no longer valid. When the 1976 Act became effective on January 1, 1978, perpetual protection for unpublished works was replaced with the same limited term of protection applied to published works. 17 U.S.C. §§ 303, 302 (1976). The district court noted the effect of the change in the law, 453 F. Supp. at 856 n.4, but applied pre-1978 copyright law. Though technically correct, perhaps, the decision is illegal.

78. *Russell v. Price*, 612 F.2d 1123 (9th Cir. 1979), *cert. denied*, 446 U.S. 952 (1980); *Classic Film Museum v. Warner Bros.*, 597 F.2d 13 (1st Cir. 1979); *Rohauer v. Killiam Shows, Inc.*, 551 F.2d 484 (2d Cir. 1977), *cert. denied*, 431 U.S. 949 (1977); *Filmvideo Releasing Corp. v. Hastings*, 509 F. Supp. 60 (S.D.N.Y. 1981), *aff'd*, 668 F.2d 91 (2d Cir. 1981); *Grove Press v. Greenleaf Publishing Co.*, 247 F. Supp. 518 (E.D.N.Y. 1965).

79. The 1976 Act allows any grant of rights by an author to be terminated between 35 and 40 years after the date of the grant. This right to terminate cannot be waived or assigned, but does descend to the author's surviving spouse and children or grandchildren. See 17 U.S.C. § 203.

An author may, of course, grant rights to another to produce a derivative work. That grant, like any other, may be terminated after 35 years in accord with section 203 of the Act. The 1976 Act

The issues are multiple: what happens to an underlying work when a derivative work falls into the public domain and vice-versa? Can the rights of the underlying work's owner be infringed by anything done with the derivative work and vice-versa? In this latter context, does it make any difference if the derivative work is in the public domain?

For these issues, a few answers exist. A derivative work's entering the public domain will not drag the underlying work with it. The wording of both the 1909 and 1976 Acts says as much;⁸⁰ the legislative history to the 1909 Act says so expressly,⁸¹ and the *Rohauer* case has interpreted the 1909 Act as saying so.⁸² There is also authority for the opposite situation: a derivative work falling into the public domain may sometimes be rescued by the copyright on the underlying work. A rescue attempt of just this sort has succeeded in at least three cases,⁸³ of which two will serve as examples—*Grove Press v. Greenleaf Publishing Co.*⁸⁴ and *Filmvideo Releasing Corp. v. Hastings*.⁸⁵

In *Grove Press*, Jean Genet, French author of *Le Journal du Voleur* (The Thief's Journal), had authorized an English-language translation of his novel. The translator failed to comply with certain United States copyright formalities and the English translation, which would be termed a derivative work under the 1976 Act, fell into the public domain. A third party, recognizing that the English version was uncopyrighted, reproduced that version for sale and was sued by the translator.

The court found that the third party's reproduction of the uncopyrighted English-language version included many of the copyrightable elements of Genet's original French work, including "the entire plot, scenes, characters and dialogue of the novel, i.e., the format and pattern."⁸⁶ Later the court used the word "pattern" several times, emphasizing that copyright in a novel protects more than just its literal wording.⁸⁷ The third party was enjoined from reproducing and selling the reproduction.⁸⁸

The effect of the decision was to take a work that anyone had a right to expect was in the public domain, out of the public domain. The decision has been criticized for failing to vindicate copyright policies (under the 1909

specifies that the recipient of the grant who produces the derivative work may continue to exploit that work after the grant's termination. 17 U.S.C. § 203(b)(1).

It is in just this context—the production of a derivative work under terms of a grant that is later terminated—that the 1976 Act clarifies the relationship of derivative to underlying works. See generally 1 M. NIMMER, *supra* note 39, § 3.07[A] at 3-23 through 3-32 (1983).

80. Act of Mar. 4, 1909, ch. 320, § 7, 35 Stat. 1077; 17 U.S.C. § 103(b) (1982).

81. See Joint Hearing before the House and Senate Committees on Patents, (March 26, 1908), (testimony of W.B. Hale), reprinted in LEGISLATIVE HISTORY OF THE 1909 COPYRIGHT ACT K-76 (1976), quoted in *Rohauer v. Killiam Shows, Inc.*, 551 F.2d 484, 489 (2d Cir. 1977).

82. *Rohauer*, 551 F.2d at 489.

83. In addition to *Grove Press* and *Filmvideo*, discussed in the accompanying text, see *Russell v. Price*, 612 F.2d 1123 (9th Cir. 1979), cert. denied, 446 U.S. 952 (1980).

84. 247 F. Supp. 518 (E.D.N.Y. 1965).

85. 509 F. Supp. 60 (S.D.N.Y. 1981), *aff'd*, 668 F.2d 91 (2d Cir. 1981).

86. *Grove Press*, 247 F. Supp. at 525.

87. Presumably, the notion of copyright protecting the pattern of a work derives from a 1945 article by Zechariah Chafee, Jr., *Reflections on the Law of Copyright: I*, 45 COLUM. L. REV. 503 (1945). Chafee used the term as an "imaginative description of what should not be imitated." *Id.* at 513.

88. *Grove Press*, 247 F. Supp. at 528.

Act), when the occasion to do so at little cost arose.⁸⁹ "Little cost" because the contrary decision would still have left Genet with the right to license all other derivative works, including any new, authorized English translations,⁹⁰ and would not have barred any member of the public from copying a work left unprotected by copyright law.

On the other hand, the decision has been supported by arguments that translations are a unique form of derivative work because they so closely resemble their underlying, original works. Other derivative works in the public domain should remain free for copying, so the argument goes, but translations rely on the original author's contribution to such an extent that they should not.⁹¹

Several cases similar to *Grove Press* have arisen, not over translations, but over the rights to motion picture films derived from novels or plays. *Rohauer*, already discussed,⁹² came out differently from *Grove Press*, with the court's refusal to find protectible elements of the *Son of the Sheik* novel infringed by the movie. The Second Circuit later almost repudiated *Rohauer*, however, in *Filmvideo Releasing Corp. v. Hastings*,⁹³ a 1981 decision.

Filmvideo arose from author Clarence Mulford's 1935 grant of motion picture rights in his Hopalong Cassidy novels. The movies were made and copyrighted, but the copyrights lapsed because of the proprietor's failure to renew them in the 1960's. The novels' copyrights were renewed and continue in effect today.

The owner of the films sought a declaratory judgment that the novels were in the public domain because of an invalid renewal, or that the movies were in the public domain and could be shown freely. The district court never questioned that films in the public domain *could* infringe a still protected novel. The only questions were whether the novels were protected, whether the films were in the public domain, and whether the films did in fact substantially copy from the novels. The district court, after reading twenty-six Hopalong Cassidy novels and viewing twenty-seven hours of Hopalong Cassidy films, answered all three questions affirmatively.⁹⁴ Some of the films were found to infringe the Cassidy books' characters, and others the storyline.⁹⁵ The district court's conclusions were affirmed by the Second Circuit,⁹⁶ with a careful distinction of *Rohauer* and a reference to that decision's "minor aberration" from the usual rule espoused in *Grove Press* and similar cases.⁹⁷

Both *Grove Press* and *Filmvideo* support the conclusion that object pro-

89. See Goldstein, *Derivative Rights and Derivative Works in Copyright*, 30 J. COPYRIGHT SOC'Y U.S.A. 209, 245-46 (1983).

90. *Id.* at 246.

91. See Nevins, *The Doctrine of Copyright Ambush: Limitations on the Free Use of Public Domain Derivative Works*, 25 ST. LOUIS U.L.J. 58, 79-80 (1981).

92. See *supra* text accompanying notes 65-67.

93. 668 F.2d 91 (2d Cir. 1981).

94. *Filmvideo Releasing Corp. v. Hastings*, 509 F. Supp. 60, 62, 65-66 (S.D.N.Y. 1981), *aff'd*, 668 F.2d 91, 92-93 (2d Cir. 1981).

95. *Id.* at 66.

96. 668 F.2d 91, 92-93 (2d Cir. 1981).

97. *Id.* at 93.

grams, though uncopyrightable, may not be copied without infringing the underlying source program. Object programs lack copyright for a reason different from the English translation of Genet's work or the Hopalong Cassidy films: object programs lack creative authorship altogether. Yet that lack is not a consequence of the source program author's fault in any sense, nor should it justify depriving the author of the most significant market by far for his works.

The *Grove Press* line of cases is based on the granting of copyright protection to an author's story line or pattern of incidents. The theory evolved in the context of novels and movies, raising the question whether there is enough of an analogy to this "pattern" notion in the context of computer programs to justify its application there.

On the surface, object programs reveal very little about their source program forebearers, as the earlier examples show.⁹⁸ Yet, all three examples do have something akin to a common "pattern": they all cause a very similar sequence of events to take place inside the computer. The "sequence of events" in a program or computer sounds a lot like the "sequence of events" in a novel. It does make sense, then, to say that object programs reflect the pattern, that is, the sequence of events, of their source programs and can be protected against copying even if they are not independently copyrightable.

A second question arises, of course, as to whether the generous judicial deference to the concept of pattern in a novel or movie ought to be carried over to far more utilitarian works like computer programs, even if that carry-over can be done conceptually. A program's sequence of events, after all, might be said to resemble the process performed by the program. Processes have not been the subject of copyright protection at least since the 1879 decision, *Baker v. Selden*.⁹⁹ The CONTU Final Report quoted the House and Senate reports on the 1976 Act as saying 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."¹⁰⁰

By itself, a reference to "expression" or "process" does not resolve the issue of which is which in a computer program. CONTU tried to resolve that issue by comparing programs to games: "One may not adopt and republish or redistribute copyrighted game rules, but the copyright owner has no power to prevent others from playing the game."¹⁰¹ The analogy breaks down with computer programs because with programs, the "rules," or program statements, "play the game" themselves.

To be sure, CONTU recognized that the process-expression distinction must ultimately be drawn by courts in individual cases.¹⁰² Yet, the Final Report contains a hint that at least one CONTU commissioner implicitly understood that object programs are different from source programs. At a

98. See *supra* note 95.

99. 101 U.S. 99 (1879).

100. CONTU Report, *supra* note 12, at 19 (emphasis added by CONTU).

101. *Id.* at 20.

102. *Id.* at 21.

CONTU hearing, Commissioner Arthur Miller questioned well-known computer expert and author Daniel McCracken:

Commissioner Miller: How many different ways are there to produce program . . . ?

Mr. McCracken: An infinite number in principle, and in practice dozens, hundreds.

Commissioner Miller: So it is comparable to the theoretically infinite number of ways of writing Hamlet?

Mr. McCracken: I believe so. It is not really true that there is a very restrictive way to write program. . . .¹⁰³

This conversation immediately suggests that there exists a *Hamlet* apart from the particular words chosen by Shakespeare, and similarly, that there exists a program apart from the particular words chosen by the programmer. In the same fashion, there must exist a *Thief's Journal* apart from the words used by Genet.

In *Grove Press*, it was just this non-verbal *Thief's Journal* that the court protected; in *Filmvideo*, it was the Hopalong Cassidy stories. The quoted dialogue appears in the CONTU Final Report in the context of a discussion of the idea-expression distinction in copyright law. CONTU seems almost certainly to have used this notion of "hundreds of ways to write a program" to draw a line: the hundred ways are protectible expressions; "the program" is the unprotectible idea or process.¹⁰⁴

Now the problem of protecting an object program as a form of expression is clearer: is not the object program itself "the program" that hundreds of source programs might be written to express? How then can it be a protectible expression itself?¹⁰⁵ One answer to this question is that the two source programming examples shown above¹⁰⁶ are so short that they are trivial—they would certainly be too short to sustain a copyright.¹⁰⁷ They might produce the same object program, but different versions of non-trivial source programs would surely not do so. Besides, once produced, an object program is fixed—otherwise we could not speak rigorously of comparing two object programs. When one speaks of *Hamlet* or "a program" apart from particular words, one speaks of something not fixed.

Fixation *makes* something an expression, though cases like *Grove Press* show that we do not confine the term, "expression," only to a single, particular fixation. Thus we come back to the conclusion that perhaps a non-trivial

103. *Id.* at 20 n.106 (emphasis added).

104. See *Apple Computer, Inc. v. Franklin Computer Corp.*, 714 F.2d 1240, 1253 (3d Cir. 1983), cert. dismissed, 104 S. Ct. 690 (1984).

105. Indeed, copyright's pervasive notion of unexpressed, hence uncopyrightable, "ideas" is a theoretical concept that may not be correct at all. Philosopher Ludwig Wittgenstein pointed out that perhaps *nothing* underlies expressions. What we take to be a variety of surface manifestations of something intangible and far down in the "levels of abstraction", see *Nichols v. Universal Pictures Corp.*, 45 F.2d 119 (2d Cir. 1930), may simply be a family of related expressions. When we see a human family whose members all look very much alike, we do not necessarily conclude that there exists a single, underlying family member—not an ancestor, but an abstract family member—from whom all this family derives. L. WITTGENSTEIN, *PHILOSOPHICAL INVESTIGATION* ¶¶ 65-67 at 31e-32e (G.E.M. Anscombe trans. 1968).

106. See *supra* note 75.

107. See CONTU Report, *supra* note 12, at 20.

object program is an expression, and that not just other source programs, but other object programs could exist that would produce the same effect when run on a computer. Only one hurdle to protection now remains: If several different object programs produce the same effect, how different can their expressions be? Different enough to sustain copyright protection?

Different object programs accomplishing the same effect can be different in the efficiency with which they accomplish it.¹⁰⁸ For example, to multiply two numbers like 156 and 6, programmers could instruct a computer to multiply: "156 \times 6." Or they could request the computer to add 156 six times: "156 + 156 + 156 + 156 + 156 + 156." Or they could even ask the reverse: "6 + 6 + 6 . . . + 6" (156 times). Although the choice of the most efficient procedure here seem obvious, other procedures for other tasks have less obvious answers.¹⁰⁹

A complicated array of choices thus faces every programmer. Once the overall function of a program is determined, the authorship of the program's source code consists largely in the selection and ordering of numerous small procedures, known in the programming trade as "algorithms."¹¹⁰ If any of the "pattern" of the source program is reflected in the object program, it is in the ordering of these procedures. Because different object programs can perform the same function, yet use different procedures to do so, each of the different object programs may legitimately be viewed as an expression.

Now the "pattern" theory looks sound: source programs contain a pattern in their selection and ordering of procedures; the object version of a program reflects that pattern; and though not independently copyrightable, an object program can be protected precisely because it copies the pattern of the source program.

A second line of reasoning supports this conclusion. Even though the programmer's surface expression in a source program is not perceptible in the program's object version, the latter version retains a substantial market value because of that expression. Careful wording of source programs, produced from a carefully chosen programming language, can greatly reduce the likelihood of programming errors. Errors can never demonstrably be eliminated, but proper programming techniques can minimize the risk of errors.¹¹¹ Those techniques have to do entirely with the programmer's selection and ordering of the words and statements of the chosen programming language¹¹²—in short, the techniques have everything to do with the programmer's expression. That expression, in turn, leads to a marketable

108. See *id.* at 20 n.106 (remarks of Comm'r Miller).

109. Frequently programmers must direct a computer to see if a given word is contained in a long list of words. For example, many word processing programs have an associated "spelling checker." The spelling program takes each word in turn from a word processing document and looks it up in a dictionary. When a computer "looks up" a word, it must compare it letter by letter with words in the dictionary to find a matching word. Should the letter-by-letter comparison be made from left to right, or from right to left?

110. J. TREMBLAY AND R. BUNT, AN INTRODUCTION TO COMPUTER SCIENCE 34-35, 63 (1979).

111. D. VAN TASSEL, PROGRAM STYLE, DESIGN, EFFICIENCY, DEBUGGING, AND TESTING, 119-120 (1981).

112. See *id.* at 117-154.

commodity: relatively error-free object programs.¹¹³

What makes different object programs performing the same function recognizably distinct, then, is both the efficiency with which they perform and their reliability or freedom from errors. In the former sense, the "pattern" of the source program expression is very much reflected in the object program; in the latter sense a valuable attribute of the programmer's expression, reliability, is also reflected. The *Grove Press* analysis seems very much to the point.

Critics of the *Grove Press* case and its pattern theory have pointed to copyright's policy of providing an incentive to authorship. They argue that incentives to authors will not diminish materially if others are allowed to copy public domain derivative works.¹¹⁴ For Genet and *Le Journal du Voleur*, all other derivative works remained his for exploitation. Furthermore, authors in the future can protect themselves from the failure by licensees like Genet's translator and publisher, or like Paramount Pictures in the *Filmvideo* case, to preserve a derivative work's copyright, by requiring indemnity clauses in their licensing contracts.¹¹⁵

Neither the incentive argument nor the licensee's indemnity argument, however, applies to the market for computer source programs. First, source programs have very little market if they cannot be translated into object programs;¹¹⁶ at least with today's preferences, a source program cannot be reworked into a movie or a television series. If the market for the object program version of a source program is lost because the object program is in the public domain, almost the entire value of the source program is lost, and hence, almost all the incentive to program authorship. Second, the translation from source to object is not performed under a licensing agreement; indemnity for a licensee's failings is therefore not possible. Indemnity makes no sense anyway because the lack of copyright protection for object programs is not a question of someone's failure to take proper steps, but rather a result of the wording of the Copyright Act.

The need for incentives, the inappropriateness of an indemnity agreement, and the theory that derivative works reflect the pattern of their underlying works all support the treatment of object programs as protectible entities.

CONCLUSION

Computer programs in a form ready for execution by a computer still pose a dilemma for copyright law, even after the 1976 Copyright Act and the 1980 Amendments. "Object programs" do not readily fit into the definitions

113. The market's concern over bug-laden programs has resulted recently in the formation of independent program testing organizations. See J. von Alten, *Getting the Software Bugs Out: Independent Testers Come to the Rescue of Consumers*, 6 INFOWORLD No. 37, at 47 (Sept. 10, 1984).

114. See, e.g., Goldstein, *supra* note 73, at 239-40, 246, 252.

115. See *id.* at 246.

116. A market does exist for instructional texts about programming and for books of unsophisticated programs in the BASIC language. The market for software is dominated, however, by expensive, complex programs sold to the business community in the form of object programs. See, e.g., *Survey of Software Users*, 225 (No. 12) PUBLISHERS' WEEKLY 50 (Mar. 23, 1984).

under the 1976 Act as independent literary works because they do not embody any originality or human authorship. Object programs look much like translations and may fit, though not exceedingly well, into the "derivative works" category. As a derivative work, an object program still cannot sustain an independent copyright, however, because it lacks originality and authorship. Nor can an object program be considered a joint work of the source program author and the author of the source-to-object translating program. Joint works result from a merger of two or more expressions; none of the translating program's expression shows up in the object programs it produces. Finally, the appealing notion that object programs are "copies" of their source programs fails to comport with the statutory definition of "copy." A copy is something from which the original work can be perceived. For programs written in a language like Basic or Cobol, the programmer's expression is lost in the translation to object code and becomes imperceptible.

Congress would doubtless prefer that the Copyright Act be read so that protection could be given to object programs without further legislative effort.¹¹⁷ Two copyright theories can be used to justify protection. First, the process of preparing an object program is so mechanical and so similar to copying that preparing and copying should be treated alike. With this approach, anyone "copying" an object program could be considered to be "preparing" one in violation of the program author's right to prepare derivative works under section 106(2) of the Act. The advantage to this theory and the pattern theory, below, is that authors may control the preparation of derivative works even if those works are not capable of sustaining copyright protection themselves.

The second copyright theory draws on *Grove Press* and similar cases to suggest that an object program reflects the copyrightable "pattern" of the source program to such an extent that the object program may not be copied even if it is itself uncopyrightable. The pattern of a source program consists of the selection and sequence of its procedures and the freedom from errors that follows from the programmer's careful choice of source program expressions.

Neither of these copyright theories seems to fall like ripe fruit from the Act. Perhaps the difficulty stems from the nature of computer programs: they are *sui generis*, both within copyright law and within human experience.

Congress could, of course, provide a kind of *sui generis* protection scheme for programs. That would not make much sense, though. When the market for goods such as computer programs is a market for easily-produced

117. "Congress probably wanted the courts to interpret the definitional provisions of the new act flexibly, so that it would cover new technologies as they appeared, rather than to interpret those provisions narrowly and so force Congress periodically to update the act." *WGN Continental Broadcasting Co. v. United Video, Inc.*, 693 F.2d 622, 627 (7th Cir. 1982). The *WGN* case interpreted a television transmission that included a separate textual display as a single audiovisual work, so that licensee television stations were not allowed to split the two signals without permission. The signal was truly a "new technology," and the court's analysis was undoubtedly correct. Computer programs were not a new technology in 1976; it is unfortunate that CONTU, and hence the Congress, did not face up to the problem of classifying object programs when the Act was still in process.

copies of an intangible work, the copyright system is about as effective a device as one could hope for. Most other countries have already adopted copyright as the protection of choice for computer programs,¹¹⁸ as have a number of United States courts. It seems late to start from scratch.

Perhaps, if the theories offered here are unsatisfactory, and if a new form of *sui generis* protection is inappropriate, a few minor touch-ups to the 1976 Act would do the job. First, Congress could simply declare that object programs are to be considered "copies" of their source programs. This change could be accomplished by adding a sentence or two to the Act's definitions of "copy" and "computer program." A short addition to the definitions would do nothing to alter the existing Act's scheme or theory of protection; the possibility that a source program could lack sufficient originality for copyright would remain; the possibility that a source program's originality might lie so much in its ideas, not its expression, that it would be uncopyrightable would also remain as it is currently.

Second, Congress could take note that the Act's section 102 list of several categories of copyrightable works is not exclusive.¹¹⁹ Computer programs could simply be removed from the "literary works" category and put into a newly-created category for "computer program works." This category would be defined to include only source programs. Congress could then add a new exclusive right to the list of rights currently given in section 106. That right would be, in the case of computer programs, to prepare, distribute, and execute on a computer any versions of the work.

This change would get to the heart of the problem with computer programs—that unauthorized copies are not just read or observed, but used for their functions. It would deviate more than the first suggested change from traditional copyright doctrine, which has focused primarily on copying and selling works, not on using them. But the deviation is no greater than that necessary, for example, to protect the use in the form of performances of plays, a "deviation" so well ingrained in copyright law that no one would argue against it.

Hardly any disinterested observer would assert that computer object programs should go unprotected entirely; the debate is whether copyright is the appropriate means of protection. When a few modest theories, or failing that, a few modest changes to the Act will suffice to accomplish that protection, the question really becomes: why *not* copyright?

118. See *Brazil Considering Sui Generis Form of Protection for Software*, 28 PAT., TRADE, & CORP. J. 482, 482-83 (B.N.A. Aug. 30, 1984) (mentioning that *sui generis* protection is disfavored by many authorities, and that copyright is already relied on by the United States, the Philippines, Hungary, and several Latin American countries, all by statute, and by France, the Netherlands, West Germany, Australia and Japan by case law). See also Pub. L. No. 98-573, The International Trade and Investment Act, October 30, 1984, section 251: "[C]opyright protection is afforded computer software by most industrialized nations including Japan, the Netherlands, France, the Federal Republic of Germany, the United Kingdom, South Africa, Hungary, Taiwan, and Australia. . . ." reprinted in COPYRIGHT LAW REPORTS CCH ¶ 20,281 at 10,743.

The Congress may not be wholly averse to *sui generis* protection schemes: just such a mechanism was recently enacted to protect certain steps in the process of creating semiconductor chips for computers and other electronic devices. See Pub. L. No. 98-620, tit. III, Nov. 8, 1984.

119. H. REP. NO. 1476, 94th Cong., 2d Sess. 53, reprinted in 1976 U.S. CODE CONG. & AD. NEWS 5659, 5666.