Published monthly
Annual subscription:
Sw.fr. 125.-
Each monthly issue:
Sw.fr. 12.-

Industrial Property

21st Year - No. 4
April 1982

Monthly Review of the
World Intellectual Property Organization (WIPO)

Contents

© WIPO 1982
Any reproduction of nfficial notes or reports, articles and translatinns nf laws or agreements published in this review is authnrized nnly with the prinr consent nf WIPO.

ISSN 0019-8625

International Unions

Nice Agreement

Ratification of the Geneva Act (1977)

GERMAN DEMOCRATIC REPUBLIC

The Government of the German Democratic Republic deposited, on March 15, 1982, its instrument of ratification of the Geneva Act of May 13, 1977, of the Nice Agreement Concerning the International Classification of Goods and Services for the Purposes of the Registration of Marks of June 15, 1957, as revised at Stockholm on July 14, 1967.

The said instrument was accompanied by the following declaration:

"The position of the German Democratic Republic on the provisions of Article 13 of the Nice Agreement as revised at Geneva, so far as the application of the Agreement to colonial and other dependent territories is concerned, is governed by the provisions of the United Nations Declaration on the Granting of Independence to Colonial Countries and Peoples (Resolution 1514 (XV) of December 14, 1960) proclaiming the necessity of bringing to a speedy and unconditional end colonialism in all its forms and manifestations." (Translation)

The Geneva Act (1977) of the said Agreement will enter into force, with respect to the German Democratic Republic, on June 23, 1982.

Nice Notification No. 53, of March 23, 1982.

Budapest Treaty (Microorganisms)

Change in Fees under Rule 12.2 of the Regulations under the Budapest Treaty

AMERICAN TYPE CULTURE COLLECTION

The following notification addressed to the Director General of WIPO by the Government of the United States of America under Rule 12.2*(a)* of the Regulations under the Budapest Treaty on the International Recognition of the Deposit of Microorganisms for the Purposes of Patent Procedure was received on March 2, 1982, and is published by the International Bureau of WIPO pursuant to Rule 12.2*(b)* of the said Regulations:

The Fee Schedule of the American Type Culture Collection as published in the April 1981 issue of *Industrial Property* is changed as follows in respect of the fee for the furnishing of a sample under Rules 11.2 and 11.3 of the said Regulations:

— selected samples furnished to high schools	US$ 10.00* (per sample)
— non-profit institutions:	
— selected samples for educational use	21.50* (per sample)
— other samples	37.00* (per sample)
— commercial firms	59.50* (per sample)

*Shipping charges are to be added to the fee itself.

[End of text of the Notification of the Government of the United States of America]

The fees set forth in the said Notification of the Government of the United States of America will apply as from the thirtieth day following the date (April 30, 1982) of the publication of the said fees in the present issue of *Industrial Property*, that is, as from May 30, 1982 (see Rule 12.2*(c)* of the Regulations under the Budapest Treaty), and will replace the fees for the furnishing of a sample under Rules 11.2 and 11.3 of the said Regulations, published in the April 1981 issue of *Industrial Property*.

Budapest Communication No. 9 (this Communication reproduces the text of the Notification of the Government of the United States of America and is the subject of Budapest Notification No. 25 of March 17, 1982).

Plant Varieties

The International Union for the Protection of New Varieties of Plants in 1981

State of the Union

In 1981, four States deposited their instruments of ratification of the Revised Act of October 23, 1978, of the International Convention for the Protection of New Varieties of Plants (hereinafter referred to as "the 1978 Act"), namely: Ireland, on May 19; Switzerland, on June 17; South Africa, on July 21; and Denmark on October 8. Those four instruments plus the instruments deposited in 1980 by New Zealand and the United States of America fulfilled the conditions for the entry into force of the 1978 Act and, pursuant to Article 33 (1) thereof, that Act entered into force on November 8, 1981. On that date Ireland, New Zealand and the United States of America became member States of the Union. The entry into force of the 1978 Act means that States may no longer accede to the UPOV Convention of December 2, 1961, as amended by the Additional Act of November 10, 1972.

The Union currently comprises the following 15 member States: Belgium, Denmark, France, Germany (Federal Republic of), Ireland, Israel, Italy, Netherlands, New Zealand, South Africa, Spain, Sweden, Switzerland, United Kingdom, United States of America.

Sessions

During 1981, the various bodies of UPOV met as described below. Unless otherwise specified, the sessions took place in Geneva.

The *Council* held its fifteenth ordinary session from November 10 to 12, 1981, under the chairmanship of Dr. W. Gfeller (Switzerland). All member States except Israel and Italy were represented. The session was also attended by observers from a number of interested non-member States, namely: Austria, Egypt, Hungary, Iran, Japan, Kenya, Mexico, Norway, Poland, Portugal. The Commission of the European Communities (CEC) and the European Free Trade Association (EFTA) were also represented by observers.

The first day of the session was devoted, for the second year running, to a symposium. The subject of the *1981 Symposium* was "Plant Breeding Activities of Government Institutes, International Centers and the Private Sector." The following lectures were given:

(i) "Plant Breeding at the French National Institute of Agronomic Research," by Mr. Jacques Huet, Head of the Department of Genetics and Plant Breeding of the French National Institute of Agricultural Research (INRA);

(ii) "CIMMYT's Crop Improvement Programs," by Dr. Ripusudan Lal Paliwal, Associate Director (Maize Program) of the International Center for the Improvement of Maize and Wheat (CIMMYT) in Mexico, and Dr. Arthur R. Klatt, Associate Director (Wheat Program) of the same Center;

(iii) "The Significance of Plant Breeding by the Private Sector," by Dr. Cornelis Mastenbroek, President of the International Association of Plant Breeders for the Protection of Plant Varieties (ASSINSEL);

(iv) "The Rice Improvement Program of the International Rice Research Institute (IRRI)," by Dr. Gurdev S. Khush, Head of the Plant Breeding Department of the International Rice Research Institute (IRRI) in Manila (Philippines).

In addition to the representatives of member and non-member States, CEC and EFTA, the Symposium was attended by some 21 representatives of institutions active in or competent for plant breeding in various member States, representatives of intergovernmental organizations (European Cooperative Programme for the Conservation and Exchange of Crop Genetic Resources (ECP/GR), Food and Agriculture Organization of the United Nations (FAO)), representatives of international non-governmental organizations (European Association for Research on Plant Breeding (EUCARPIA), International Association for the Protection of Industrial Property (IAPIP), International Association of Horticultural Producers (AIPH), International Association of Plant Breeders for the Protection of Plant Varieties (ASSINSEL), International Federation of the Seed Trade (FIS)) and representatives of international breeding centers supported by the Consultative Group on International Agricultural Research (CGIAR) (International Center for Agricultural Research in the Dry Areas (ICARDA) in Beirut (Lebanon) and CIMMYT and IRRI, the centers which provided lecturers).

The Symposium was concluded by a panel discussion. The presence of a number of experts from CGIAR-supported international breeding centers provided a valuable opportunity to expand the dialogue between those experts and representatives of UPOV member States regarding the policies to be adopted by the centers on the one hand and by UPOV and the plant variety protection offices of UPOV member States on the other.

Records of the proceedings of the Symposium will form the subject of a special publication.

The following main decisions were taken by the *Council* at its fifteenth ordinary session:

(i) the report of the Secretary-General on the activities of the Union in 1980 and the first ten months of 1981, the report on his management and the financial situation of the Union in 1980, and the accounts of the Union for 1980, were approved;

(ii) the program and budget for 1982 was examined and approved;

(iii) the reports on the progress made by the various committees and technical working parties, including their plans for future work, were approved; in that connection, the recommendation of the Technical Committee that the Technical Working Party for Forest Trees be incorporated in the Technical Working Party for Ornamental Plants was adopted;

(iv) the recommendation of the Consultative Committee that an information meeting be held in 1982 with international non-governmental organizations was adopted;

(v) the recommendation of the Consultative Committee that the 1982 Symposium be devoted to the technical and legal aspects of genetic engineering and of cell, meristem and tissue cultures was adopted;

(vi) the following officers were elected, for a term of three years expiring at the end of the eighteenth ordinary session of the Council (1984):

(a) Mr. J. Rigot (Belgium) was elected Vice President of the Council;

(b) Dr. G. Fuchs (Federal Republic of Germany) was elected Chairman of the Technical Working Party for Agricultural Crops;

(c) Dr. G.S. Bredell (South Africa) was elected Chairman of the Technical Working Party for Fruit Crops;

(d) Mrs. U. Löscher (Federal Republic of Germany) was elected Chairman of the Technical Working Party for Ornamental Plants and Forest Trees;

(e) Mr. F. Schneider (Netherlands) was elected Chairman of the Technical Working Party for Vegetables.

The *Consultative Committee* held its twenty-third session on May 6 and 8, 1981, and its twenty-fourth session on November 9 and 12, 1981, both under the chairmanship of Dr. W. Gfeller (Switzerland). All member States were represented at the twenty-third session and all except Israel and Italy at the twenty-fourth session. The sessions were devoted mainly to the preparation of the fifteenth ordinary session of the Council.

The *Administrative and Legal Committee* held its seventh session on May 6 and 7, 1981, and its eighth session from October 12 to 14, 1981, both under the chairmanship of Mr. P.W. Murphy (United Kingdom). All member States were represented at the seventh session and, with the exception of Italy, at the eighth session. Both sessions were attended by observers from Ireland, Japan, the United States of America and the CEC; in addition, observers from Canada and New Zealand attended the eighth session.

As in 1980, the Committee gave priority to the matter of harmonization of national legislation and practice. At its seventh session, the Committee noted the intentions of member States as regards amendment of their legislation, both in relation to ratification of the 1978 Act and in general. At its eighth session, it examined three specific items of plant variety protection law:

(i) having examined the question of extending the scope of protection beyond the minimum provided for in Article 5(1) of the Convention, the Committee felt that, apart from the case of plantlets, extension of protection should be envisaged only for ornamental plants and fruit crops and that the aim of the extension was to safeguard the interests both of breeders and of those producers who paid royalties and suffered the competition of products not having to bear royalties; it was generally agreed that protection should be extended to the propagation of plants intended for the commercial production of the final product (cut flowers or fruit), but some delegations expressed reservations as to the protection of the final product itself in the case of ornamental plants; the Committee invited the small number of member States that had based their domestic legislation on an interpretation of Article 5(1) of the Convention reducing considerably the extent of protection, particularly as regards "adult" plants sold to the final user, to re-examine their point of view;

(ii) having examined the optional provision contained in the second sentence of Article 5(4) of the Convention, whereby member States granting a more extensive right—extending in particular to the marketed product—may limit the benefit of it to the nationals of member States of the Union granting an identical right and to natural and legal persons resident or having their registered office in any of those States, the Committee noted the drawbacks that could arise from the use of that provision;

(iii) having been informed that one member State was considering the need to exclude parent hybrids from protection, the Committee noted that the fact giving rise to such consideration was that a person could block or disturb the creation and marketing of a certain number of commercial hybrids by obtaining protection for the intermediate or parent hybrids necessary for the production of seed of the commercial hybrids.

At both sessions, the Committee investigated ways of revising and improving the much discussed Guidelines for Variety Denominations, originally adopted by the Council at its seventh ordinary session (1973). The Committee began to give consideration to replacing the Guidelines by a set of recommendations on the interpretation of the revised version of Article 13 of the Convention, now in force for member States bound by the 1978 Act. Such recommendations would be illus-

trated by examples of designations which are or are not suitable as variety denominations. At its eighth session, the Committee reached agreement on two principles:

(i) combinations of letters and figures—in that order—should be accepted in the case of species, such as maize and sorghum, for which this type of denomination is an established international practice; this should also apply to series of denominations including the same alphabetical component, but it should be understood that no breeder would have an exclusive right to such a component;

(ii) in the case of a series of denominations based on a fancy name and applied to a family of varieties developed generally as a result of mutations, any new denomination in the series should not represent a simplification of the earlier denominations.

In last year's report (see *Industrial Property*, 1981, p. 97), reference was made to the intention of the Committee to examine the possibility of developing a system of more far-reaching cooperation. At its eighth session the Committee confirmed its intention to resume, at the appropriate time, its study of a system going beyond the bounds of variety examination.

The *Technical Committee* held its seventeenth session from October 14 to 16, 1981, under the chairmanship of Mr. C. Hutin (France). All member States except Italy were represented. The session was also attended by observers from Canada, Ireland, Japan and New Zealand.

The main results of the session were as follows:

(i) The Committee adopted seven revised Test Guidelines submitted:

(a) by the Technical Working Party for Agricultural Crops—for Wheat (TG/3/8), for Barley (TG/19/7) and for Oats (TG/20/7);

(b) by the Technical Working Party for Ornamental Plants—for Euphorbia fulgens (TG/10/4) and for Poinsettia (TG/24/5);

(c) by the Technical Working Party for Vegetables—for Peas (TG/7/4), revised in conjunction with the Technical Working Party for Agricultural Crops, and for Lettuce (TG/13/4).

(ii) The Committee noted some problems that had arisen in connection with the examination of new varieties, especially in species in which mutations occurred rather easily. It discussed in detail the standards that should be applied in examining new varieties for distinctness. It recalled in this respect that a variety had—according to the Convention and the national laws based on it—to be clearly distinguishable by one or more important characteristics from any other variety commonly known at the time when protection was applied for. It underlined the fact that the examination for distinctness conducted by the plant variety protection authorities of member States had to go beyond checking merely whether two samples were identical or not, and the fact that methods used for identification purposes—that is to say, for determining the variety to which a sample belonged—were not always sufficient for the examination for distinctness. To be used for *identification purposes* a method had to fulfill several technical requirements. It had to be *capable of standardization* and should lead to the establishment of *significant differences* which were *consistent* and *repeatable*. Such a method might not, however, be acceptable on its own for *establishing distinctness*. Account had to be taken of the fact that the variety had to be distinguishable by the expression of an *important* characteristic, and that it must be *clearly* distinguishable. It was the Committee's view that decisions in this area should be taken species by species, bearing in mind the state of development of breeding. They should not be taken in the light of technical aspects alone. The Committee thought that this line of thinking should in particular be followed when deciding whether characteristics which could only be observed by certain sophisticated methods such as electrophoresis or various other chemical analyses were acceptable. The same would be true for the acceptance of certain disease resistances as characteristics for distinctness purposes.

(iii) As far as the question of minimum distances between varieties is concerned—a question closely connected with the above-mentioned problem of the standards of examination for distinctness—the Committee took the view that, before any decision was taken, a discussion should be held with representatives of breeders and growers.

(iv) The Committee agreed on a procedure for the exchange between Offices of member States of lists of varieties under test.

As in previous years, the Committee supervised the work of the Technical Working Parties, giving guidance on a number of questions raised by them and instructing them on the main aspects of their future work.

The *Technical Working Party for Agricultural Crops* held its tenth session in Edinburgh (United Kingdom) from June 23 to 25, 1981, under the chairmanship of Miss Jutta Rasmussen (Denmark). In addition to its work on the revised Test Guidelines adopted by the Technical Committee, the Working Party completed the preparation of first drafts of Test Guidelines for Soya Bean and for Sunflower for submission to the professional organizations for comment.

The *Technical Working Party for Vegetables* held its fourteenth session in Wädenswil (Switzerland) from September 8 to 10, 1981, under the chairmanship of Mr. J. Brossier (France). In addition to its work on the revised Test Guidelines adopted by the Technical Committee, the Working Party completed the preparation of first drafts of revised Test Guidelines for French Beans and of Test Guidelines for Celery for submission to the professional organizations for comment.

The *Technical Working Party for Fruit Crops* held its twelfth session in Wageningen (Netherlands) from September 23 to 25, 1981, under the chairmanship of Mr. A. Berning (Federal Republic of Germany). It prepared first drafts of Test Guidelines for Citrus and for Japanese Plum and of revised Test Guidelines for Apple for submission to the professional organizations for comment.

The *Technical Working Party for Ornamental Plants* held its fourteenth session at Antibes (France) from October 6 to 8, 1981, under the chairmanship of Mr. A.J. George (United Kingdom). In addition to its work on the revised Test Guidelines adopted by the Technical Committee, the Working Party began discussing working papers on Test Guidelines for Narcissi and on revised Test Guidelines for Carnation. In both cases, however, some further discussion will be required during its next session. It also took note of a report on a Chrysanthemum workshop held in Hoddesdon (United Kingdom) on November 4 and 5, 1980.

During the year under review each of the Technical Working Parties discussed several items connected with the examination of varieties and the implementation by the national plant variety protection offices of the various guidelines for the conduct of tests for distinctness, homogeneity and stability. In addition to the questions of standards for distinctness and minimum distances referred to above in the report on the seventeenth session of the Technical Committee, the other principal questions considered were: color characteristics and their measurement; the maximum number of secondary off-types, such as mutations occurring during the examination of a variety, beyond which a variety is to be considered as not being sufficiently homogeneous; the financial and practical problems associated with the maintenance of reference collections for certain species, in particular among the species that are vegetatively propagated; the standardization of tests for resistance to pests and diseases. It is envisaged that the working parties will continue their discussions on the above items in 1982 and will give particular attention to a general revision of the way in which characteristics are selected for inclusion in the various test guidelines.

Contacts with States and Organizations

During 1981, the Vice Secretary-General of UPOV established contacts with government representatives of Mexico and paid a visit to the Secretariat for Foreign Relations in Mexico City. He had further contacts with the Kenyan authorities during a visit to Nairobi. The Secretary-General and the Vice Secretary-General visited the new premises of the *Bundessortenamt* (German Federal Plant Varieties Office), in Hanover, and one of the variety testing stations attached to that Office. The President of the Council and the Vice Secretary-General paid a courtesy call to the newly appointed Director of the *Station fédérale de recherches agronomiques de Changins* (Swiss Federal Agricultural Research Station).

UPOV was represented at a tripartite meeting of experts from UPOV, the International Vine and Wine Office (IWO) and the International Board for Plant Genetic Resources (IBPGR), held in February at Colmar, France, to discuss the preparation of a "Universal List of Characteristics of the Genus *Vitis*"; at the annual Congresses of the International Association of Plant Breeders for the Protection of Plant Varieties (ASSINSEL) and of the International Federation of the Seed Trade (FIS), both held in May in Acapulco, Mexico; at the FAO/SIDA Technical Conference on Improved Seed Production, held in June at Nairobi, Kenya; at the *Festakt* (commemoration) and the lecture and discussion meeting, held in September in Vienna, Austria, to mark the centenary of the Austrian *Bundesanstalt für Pflanzenbau und Samenprüfung* (Federal Institute for Plant Production and Seed Testing); at the Thirty-Third Congress of the International Association of Horticultural Producers (AIPH), held in September in Taormina, Italy; and at the second meeting of the Governing Board of the European Co-operative Programme for the Conservation and Exchange of Crop Genetic Resources (ECP/GR), held in December 1981 in Geneva, Switzerland.

The Office of the Union was informed that a new association, entitled "Japan Association for the Protection and Development of Plant Varieties" (JAPDPV), has been established in Japan, one of the signatory States of the 1978 Act. JAPDPV is mainly concerned with the protection and development of plant varieties and is composed of persons representing business circles, science and technology, and the legal profession.

Publications

In 1981, the Office of the Union published the *Records of the 1978 Geneva Diplomatic Conference on the Revision of the International Convention for the Protection of New Varieties of Plants*, in English and German (UPOV publications 337 (E) and 337 (G), respectively); three issues of the *UPOV Newsletter* (which, from 1982, will be called *"Plant Variety Protection—Gazette and Newsletter of the International Union for the Protection of New Varieties of Plants"*); a brochure containing the Dutch text of the UPOV Convention of 1961, of the Additional Act of 1972 and of the Revised Act of 1978 (UPOV publication 293 (D)); and the *Records of the 1980 UPOV Symposium on "The Use of Genetic Resources in the Plant Kingdom,"* in English, French, German and Spanish (UPOV publications 336 (E), (F), (G) and (S), respectively).

General Studies

The Legal Protection of Computer Software and Computer-Related Innovations in the United States*

R. H. STERN**

The present state of legal protection available for computer software in the United States is uncertain and unsatisfactory. As a result, proprietors of software must often resort to complex, cumbersome arrangements to license their software and protect it from appropriation by others; sometimes, they do not avoid gaps in legal protection, through which their power to exclude others from using their software innovations slips away. The protection available under United States law for machine systems or industrial processes utilizing particular software may be somewhat more favorable, but it too is far from satisfactory. Accordingly, there appears to be a need for legislation specially directed to software and according its proprietors kinds of protection designed to meet the special needs of the subject matter and of the users and creators of software.

Under present United States law, four systems of civil protection for software exist: copyright, patent, contract, and trade secret.[1] This paper addresses the availability of these protections, and their practical shortcomings. The paper then considers an alternative form of *sui generis* protection for software, which could overcome these shortcomings.

I. Background

In the discussion that follows, it will become apparent that the law treats different aspects of software differently. Accordingly, certain definitions are used to facilitate analysis. Moreover, understanding the rationale for such distinctions requires some understanding of their physical and technological background.

Software. The term "software" will be used generically to include algorithms and any type of computer program or programming.[2]

Algorithm. The term "algorithm" will be used to refer to a procedure or set of steps for solving a problem. Ordinarily, but not necessarily, the solution is by means of a digital computer. An algorithm is expressed in ordinary or "natural" language (English, French, etc.), but the expression may contain mathematical formulas. An algorithm may also be expressed diagrammatically as a "flow chart."

Program. The term "program" will be used to refer to a particular procedure for operating a digital computer (or machine system incorporating a digital computer) to accomplish a particular result.[3] Any algorithm *A* can be embodied or particularized in many different programs *P1, P2, P3...* designed to carry out the same algorithm.[4] Usually, a program will have many more steps in it than

* © 1981 by Richard H. Stern.

This is the first of a two-part study. The second part will be published in the May issue. This study was submitted for publication in August 1981 and analyzes the state of the law up to that date.

** Partner, Baker & Hostetler, Washington, D.C., U.S.A.; Chief, Intellectual Property Section, Antitrust Division, U.S. Department of Justice, 1970-78.

[1] Trademark protection is also available for software under United States law, but it concerns only the apparent source of the software, and protects against competitors' marketing the software by falsely suggesting that they are the original source. Generally, trademark protection will not afford relief against software copying, as such, without any deception of customers as to source.

Unfair competition law may in some parts of the United States afford direct protection against copying, under the "misappropriation" doctrine emanating from the decision in *International News Syndicate v. Associated Press*, 248 U.S. 215 (1918). See, *e.g., Data Cash Systems, Inc. v. JS&A Group, Inc.*, 480 F. Supp. 1063, 1070-71 (N.D. Ill. 1979), *aff'd on other grounds*, 628 F.2d 1038 (7th Cir. 1980); Case Comment, 2 Eur. Int. Prop. Rev. 25 (1980). The doctrine is not generally applied, however, throughout the United States. Moreover, in its wider applications the misappropriation doctrine may conflict with and therefore be "preempted" by the federal patent and copyright laws, so that application of the doctrine in the face of such federal laws is prohibited. See *Compco Corp. v. Day-Brite Lighting, Inc.*, 376 U.S. 234 (1964); *Sears, Roebuck & Co. v. Stiffel Co.*, 376 U.S. 225 (1964); 17 U.S.C. § 301. See also *Synercom Technology, Inc. v. University Computing Co.*, 474 F. Supp. 37 (N.D. Tex. 1979).

[2] The term "software" is not applied here to documentation or explanatory material relating to programs. Compare WIPO, *Model Provisions on the Protection of Computer Software* (1978) (hereafter cited "WIPO, *Model Provisions*"), pp. 9 to 12. Under United States law such material is treated in all material respects like a book or other literary work. See notes 12 and 13 *infra* and accompanying text. This paper does not consider the legal status of such documentation. It also does not deal with data bases, apart from software used to access data stored in them. Again, data bases are treated like books.

[3] The recent Computer Software Copyright Act of 1980, Pub. L. 96-517, 94 Stat. 3015, amending 17 U.S.C. § 101, defined a computer program as "a set of statements or instructions to be used directly or indirectly in a computer in order to bring about a certain result." In WIPO, *Model Provisions*, "computer program" was defined as "a set of instructions capable, when incorporated in a machine-readable medium, of causing a machine having information-processing capabilities to indicate, perform or achieve a particular function, task or result." *Id.* at 9-11.

[4] An algorithm may be considered as a list or designation of a sequence of operations that if successively performed on data will result in the availability of further data derived from the original data in accordance with a predetermined scheme of derivation. A program may be considered as a list or designation of a sequence of operations that, if performed on data representations made available to a computer, will result in the availability of further data representations derived from the original data representations in accordance with a predetermined scheme of derivation. Data representations are methods of expressing data in computer-usable form, such as electronic

the algorithm it embodies, because program language (the artificial language in which a program is written) has syntax rules that usually require several statements where ordinary or "natural" language would use one statement. A long, complex program *P* may embody several algorithms *A1, A2, A3*....

The characteristic power of the computer in processing data results from the ability of computers to perform reiterations thousands of times faster than a person could. This power is utilized, typically, in algorithms and programs by including steps in which a result thus far achieved or an input variable is compared with some reference datum (*e.g.*, the computer ascertains whether the number I is less than 10) and, depending on the answer, the machine either reiterates a prior sequence of steps or else goes on to a new step (*e.g.*, if I is less than 10, the next step is to go back to the twentieth previous step and reiterate the intervening 20 steps, but if I is 10 or greater, the next step is to move forward to a further step). Such branching paths (usually with only two forks) that loop back on themselves are characteristic of algorithms, flow charts illustrating them, and programs.

The categories of "natural language" (*i.e.*, ordinary speech), which are used in an algorithm to describe the mental steps by which a person would, at least in principle, perform a calculation, differ vastly in number and kind from the categories of data processing by machine operation. The categories of natural language are so numerous as to seem infinite, are indefinite in meaning both because they are vague and because they mean different things in different contexts, and are only imperfectly understood by their users. The categories describing machine operation ultimately reduce to two possible signal levels or electronic states ("0" and "1") at particular places within the computer. The level or state can be information in itself or can control the routing (pass or stop) of other information through the particular place.

It is common to go from the natural language categories of an algorithm to the other end of this spectrum, the categories of machine operation, by descending through a hierarchy of several different kinds of program. The hierarchy includes "source programs," "assembly programs," and "object programs," described below. It is far easier, at least conceptually, to take these several smaller transitions successively than to try to go in one step from (1) natural language intelligible to humans to (2) directions for placing at locations within a computer 0/1 signal levels that in some way correlate to the intelligible, natural language. This multi-step procedure works well, from a technological standpoint, but, as will appear, the categories of United States law for dealing with rights to speech and those for dealing with rights in machinery do not seem to bridge the transitions at all as effectively.

Source Programs. The term "source program" will be used to mean a program (as that term was defined above) written in a "high level" computer programming language such as FORTRAN, BASIC, or COBOL. (A statement in such a language is said to be in "source code.") A source program is thus a series of formalized source code statements ("instructions") for operation of the computer, or instructions directed to the person using the computer, stating that a series of things is to be done to data made available to the computer.[5] The term "high level" language just used to describe source code means that such languages as BASIC are close to ordinary or "natural" English. Indeed, BASIC may be regarded as an artificial dialect of English. It is most common to write programs in source code, at the present time. It is not common, however, to market software in source program form, in large part because it is too easy to appropriate a source program, or its concept, without detection or remedy.

Assembly Code. The next level down in the hierarchy of programs is a program (usually called an "assembly program") in which an instruction of the algorithm or source code type, *add* $x + y$, is replaced by a series of instructions concerning movement of signals from one location within the computer to another. Assembly programs are written in a conventional terminology known as "assembly code" or an "assembly language."[6] It used to be quite common to write programs directly in assembly code, but it is less common now, although

signals, holes in punched cards, and magnetized portions of tapes or discs. See *Gottschalk v. Benson*, 409 U.S. 63, 65 (1972).

One difference between an algorithm and a program is in their respective generality. See *id.* The operations in an algorithm are usually concisely described in terms of conventional human mental steps for doing calculation or the like with data. The operations in the program are described in much less concise and more specific wording, in terms of computer-oriented steps (which are usually based on the concept of moving a signal from one place in a computer to another place). A program is usually much longer than an algorithm.

There is a new form of software—programs in "query language"—that is even more concise and like natural English than an algorithm. Query used in preparing reports. See note 6 *infra*.

[5] A typical BASIC instruction to the user of the computer can be of the form "INPUT X" which means that the computer will print out a *?* on the display, whereupon the user should type the correct value of variable *X* on the keyboard. Such an instruction often may be preceded by an instruction to the computer such as "PRINT 'WHAT IS RADIUS, IN INCHES'," which will cause the computer to print out "WHAT IS RADIUS, IN INCHES" before the *?* and to treat whatever number the user types after the *?* as the value of *X*. Thus such an instruction to the user may be considered an instruction to the computer to print out an instruction to the user.

Other typical BASIC instructions to the computer can be of the form "GOTO 130," meaning that the computer should carry out the instruction on line 130 of the program; "IF I/10 THEN GOTO 130," meaning if the value of variable I is less than 10, the computer should carry out the instruction on line 130, but otherwise it should do the next listed step; or "PRINT 'ANSWER IS'; X + Y," meaning that the computer is to calculate the sum of variables X and Y and print out the message "ANSWER IS _____," where the foregoing sum is printed in the blank.

[6] For example, such a direction would be one to take the data (signal) stored in memory location *T* and place it in register location *R*, after erasing whatever other data was stored at location *R*. Other typical

programs for other than data processing purposes (such as for video games) are commonly still written directly in assembly code.

Different computers can utilize the same source program, but their internal design and wiring are likely to require use of different assembly languages, and therefore require different assembly programs derived from the source program. Thus, assembly programs for the IBM 370 series of computers may be written in an assembly language called BAL, while those for the DEC PDP-11 series of computers may be written in the assembly language called MARCO-11. Software is sometimes marketed in assembly program form. There has thus far not been significant litigation involving assembly programs.

Object Code. The terms "object code" or "object program" will be used to refer to a mechanical counterpart of a source or assembly program. Object code is directly usable in a machine, and is not written at all (in the ordinary sense of that word), but is embodied in magnetic tape, disks, or another physical device (such as a read-only-memory or "ROM").[7]

Object code does not correspond to the source code to which it relates, on a simple and straight-forward basis. Object code is not simply another artificial dialect of English. Nor is it a translation of source code in the sense that one translates a book from French to English, or even translates a source program from FORTRAN to BASIC.[8] In fact, it cannot be understood by human beings, because if read out it would appear as a meaningless sequence of electronic signals of "0" and "1" value.[9] An object program is a collection of object code that, when made available to a computer for which it was prepared, will directly enable the computer to execute the program of whose source code version the object program is a counterpart.[10]

Compilation. The transformation of source code into object code may be accomplished in various ways. Typically, it is done in one or more stages, in the course of which the set of statements progresses from a high level language (*e.g.*, BASIC) through one or more successively lower language levels (*i.e.*, levels more and more removed from ordinary or natural English and less and less intelligible). The process, sometimes termed "compilation," can be accomplished by a computer and appropriate programming. A compiler program will be uniquely usable for a particular language (*e.g.*, FORTRAN), processor, and program storage device (*e.g.*, a particular model of ROM). Many different compiler programs exist, moreover, for particular such combinations.[11]

In the course of compilation, typically each source code statement will be replaced by one or more lower level statements, between which "linking statements" may be inserted. Some statements in the source code version may disappear. The ultimate product of the process will be a set of 1/0 signal level sequences (object code). These may be stored in locations on a tape or other memory storage device. When the device using the program is self-contained, the storage medium will often be a ROM, a postage-stamp size device containing thousands of chemically printed switches within it. The pattern of open and closed switches (*0*'s and *1*'s) embodies the object code. When a large, multi-purpose computer is involved, the storage medium is more likely to be tapes or disks. On these, the pattern of unmagnetized and magnetized zones on the tape or disk embodies the object code.

---

assembly code statements are of the form: "EXA 1010" meaning that the computer should execute the instruction at address 1010; "STO" meaning that the computer should send the signal to the output.

The same program will be much longer in assembly code form than source code form. An example of a program to generate a report of account balances for a list of customers is discussed in *Business Week*, September 1, 1980, at p. 48. The program has over 3,000 lines in assembly code and over 600 lines in COBOL source code. In "query language," see note 4 *supra*, the program is one sentence long ("I want a report for business type 17 in order by branch, showing account number, name, and both current and prior balances.").

[7] Source code can also be embodied in tape or other devices, and need not be printed on paper to be eventually retrievable and understandable. Source code, however, is not directly usable in a machine, for it must first be transformed into a machine-usable language.

[8] Object code has syntax rules that require a program in object code to contain many more statements than does the program in source language. There is no simple 1:1 or 1:5, etc., relationship between the source code and object code statements. In transforming source code to object code, some source code statements may wholly disappear, while some object code statements may have no counterpart in the source code version.

[9] See *Data Cash*, note 1 *supra*, 480 F. Supp. at 1065.

[10] A particular source program will have many object program counterparts. Generally, one could prepare at least one object program for each different type of computer, and usually there will be several different object programs prepared for each different type of computer on which the program is actually used.

[11] Two very common methods of developing object code are by use of "compiler" and "assembler" programs. A compiler program transforms a program in source code into object code, so that as far as the user is concerned the input is source code and the output is object code. This does not mean that the computer accomplishes this transformation in a single pass.

An assembler program transforms a program from assembly code to object code. This means that the program had to be written in assembly code in the first place (which is feasible and used to be common practice) or else transformed from source code to assembly code. Assembly language is a lower level language than source code. Particular assembly code versions of a single source program will be specific for different types of computer (*e.g.*, DEC series PDP-11 or IBM series 370).

II. Copyright

Copyright protection for particular forms of software is readily available. But the scope of the protection is limited essentially to source programs.

A. What May Be Copyrighted

Source programs are written in a language readable by human beings, such as FORTRAN or BASIC, and contain an intelligible message. They are therefore generally agreed to be "writings" or "works of authorship,"[12] are protectable under United States law as literary works,[13] and will be accepted for registration in the Copyright Office.[14] *A fortiori*, a written description of an algorithm can be deemed a literary work and thus protected by copyright ("copyrightable").[15]

On the other hand, it is clear that a machine system utilizing particular software is not copyrightable under United States law, even if the software *per se* were copyrightable. By the same token, a programmed memory device is not copyrightable, as such. Whether the object code "message" within such a device is copyrightable, as such, is a difficult, highly controversial question. Most probably, object code and object programs are not copyrightable, as such, because they are virtually unintelligible to human readers.[16]

The phrase "copyrightable as such" has been used here, because a potential distinction exists between: (1) having a copyright in X and protecting X under the copyright laws, and (2) having a copyright in Y and protecting X because of its relationship to Y. The conclusion that object code is not copyrightable as such does not transparently require the further conclusion that duplication of object code cannot be the infringement of some other thing considered to be a copyrightable writing, such as a source program. As will appear, however, the two conclusions are actually linked.

B. Scope of Copyright Protection

The scope of protection afforded copyright owners, in terms of what acts by others are prohibited as copyright infringements, is set forth in section 106 of the current copyright law.[17] That section provides that the owner of copyright has the exclusive rights to do or authorize the following things, among others:

(1) reproduction of the copyrighted work "in copies";
(2) preparation of "derivative works based upon" the copyrighted work;
(3) distribution of "copies" of the copyrighted work, by sale, rental, lending, etc.

Unless the conduct of the alleged infringer falls within one of these categories, the owner cannot secure copyright law relief against the alleged infringer.

As indicated above, source programs and algorithms are copyrightable as literary works. But the scope of protection afforded such works against infringement is limited, because copyright law prohibits only "copying" in a nearly literal sense of that term. First, only the particular literary expression of an idea is protected by copyright; the idea itself, embodied in the expression, is not protected.[18] Second, actionable copying is found

[12] The United States Constitution, Art. 1, § 8, cl. 8, authorizes Congress to pass laws to secure to "authors" a limited exclusive right to their "writings." The United States copyright laws no longer use the term "writings" to describe what is protected under copyright law. Instead, they now use the term "original works of authorship" (17 U.S.C. § 102), but the Constitution nonetheless limits the scope of the statute to "writings." See *Graham v. John Deere Co.*, 383 U.S. 1, 5 (1966) (discussing same clause of Constitution from patent standpoint). A non-writing cannot be copyrighted. See *The Trademark Cases*, 100 U.S. 82 (1879). In that decision, the Supreme Court held that Congress cannot regulate trademarks under the Patent-Copyright Clause. As a result, it was necessary to pass a new trademark law based on the Commerce Clause of the Constitution.

[13] 17 U.S.C. § 101 defines "literary work" to include "works" (undefined) expressed in words, numbers, or "other" verbal or numerical "symbols," in books, tapes, discs, or cards. The House Committee Report on the 1976 Copyright Act, H.R. Rep. No. 94-1496, 94th Cong., 2d Sess. (1976), states that literary works include "computer programs." *Id.* at 54. The Senate Report, S. Rep. No. 94-473, 94th Cong., 1st Sess. (1975), also indicates that computer programs are copyrightable as literary works. *Id.* at 54. The Computer Software Copyright Act of 1980, note 3 *supra*, makes it clear that source programs are copyrightable.

[14] Since May 1964 the Copyright Office has accepted source programs for deposit and registration as literary works. Copyright Office, Announcement of May 19, 1964, "Copyright Registration for Computer Programs," reprinted in 11 Bull. Copr. Soc. U.S.A. 361 (1964); Cary, "Copyright Registration and Computer Programs," 11 Bull. Copr. Soc. U.S.A. 362 (1964). In the Announcement, *supra*, the Copyright Office indicated some doubt as to "whether a program as such is the 'writing of an author'" and whether object code is a "copy" that can be registered. Nevertheless, it determined to resolve doubt in favor of registrability, because doing so would facilitate access to the courts for proprietors of software and permit the courts to resolve the copyrightability issue. *Id.* The Office therefore determined that it would accept copies of programs in human-intelligible form for registration, but when publication (*i.e.*, dissemination) "was in a form that cannot be perceived visually or read," a printout or the like would have to be deposited for registration. *Id.*

Most authorities now assume that source programs are protected by copyright regardless of the coding medium (*e.g.*, "hard" copy, printed on paper; disk; tape, or ROM). An exception to this is Pope and Pope, "Protection of Proprietary Interests in Computer Software," 30 Ala. L. Rev. 527, 546 (1979) (duplication of punched cards or magnetic tape containing source code is not making an "infringing copy" of source program).

[15] A flow chart expressing or illustrating an algorithm may also be copyrighted, as a "graphic" work. That term includes charts, technical drawings, and diagrams. See 17 U.S.C. § 101.

[16] See *Data Cash*, note 1 *supra*, 480 F. Supp. at 1065, 1066-67 n. 4. Physical objects are copyrightable only when they are paintings, sculptures, or other works of art. Physical objects are not copyrightable insofar as they are utilitarian. See 17 U.S.C. §§ 101 (definition of "pictorial, graphic, and sculptural works"); 102(a)(5). See also *Esquire, Inc. v. Ringer*, 591 F.2d 796 (D.C. Cir. 1978).

Even if the United States copyright statute were interpretable as extending to programmed devices or object programs stored in them, it is questionable that they are "writings" within the meaning of the Constitution. See note 12 *supra*. Accordingly, the statute as so interpreted would seem to be an invalid exercise of legislative power. See *The Trademark Cases*, note 12 *supra*.

[17] 17 U.S.C. § 106. In 17 U.S.C. § 501, infringement is defined as violation of the rights conferred in section 106.

[18] See generally *Baker v. Selden*, 101 U.S. 99 (1879). The law in the U.K. is the same. See, *e.g.*, *LB Plastics Ltd. v. Swish Products Ltd.*, [1979] F.S.R. 145 (H.L.).

only when the alleged infringer has access to a work and then duplicates the work or tracks its form of expression so closely in another writing that the second work is "substantially similar" to the first.

Algorithms and Concepts. Consequently, copyright law has several major shortcomings as a method of protecting software. The first of the foregoing principles prevents any effective protection of algorithms. Restating an algorithm in other words will ordinarily not be a copyright infringement. Using an algorithm to create a source program is not a copying of the algorithm. To be sure, a paraphrase of a work may be deemed a copy. But when the idea embodied in the algorithm can be expressed only in words that amount to a paraphrase of the original, the paraphrase will not be an infringement, because the idea and expression have merged. The wording of a book describing the laws of relativity or mass-energy equivalence can be protected from infringement by copyright; but that copyright will not prevent another person from expressing the same ideas in other words, and it definitely cannot protect against others' use or discussion of the equation $E = mc^2$.[19] For the same reasons, a second source program, based on the ideas embodied in a copyrighted first source program will not be a "copy" of the first source program unless it also comes very close to it in expression.

Proving Copying. Another shortcoming of the United States copyright law is potential difficulty in proving access and substantial similarity in the case of source programs. A translation of a work, from English to French, or in the case of a source program from FORTRAN to COBOL, is an infringement,[20] either as a copying or as the preparation of a derivative work. When the translation is quite close, there may be little problem in inferring access and demonstrating copying of the protected expression, and thus establishing infringement. Unfortunately for software proprietors, however, skilled programmers can usually rewrite a source program so that the second one does not resemble the first one, at least to the casual or inexpert observer.[21] When this occurs, it may be extremely difficult to detect and then prove copyright infringement.

Object Code. Finally, object code cannot effectively be protected under present United States copyright law. As will appear, copyright law does not protect object code on the basis that it is a "copy" or derivative of something itself copyrightable. Object code is probably not a copy of a source program, under the 1909, 1976, or 1980 Copyright Acts. Nor is it a derivative work derived from a source program. Major portions of this paper are directed to the problems in protecting object code from appropriation.

Data Cash. The severe problems encountered in trying to secure copyright relief against appropriation of software, particularly object code, are illustrated by the trial court's decision in the *Data Cash* case.[22] Data Cash had hired a programming consultant to develop a computer program for use in a hand-held-calculator type of chess game (known as "CompuChess"). The program was stored as object code in a read-only-memory ("ROM") incorporated into Data Cash's product. About a year after Data Cash began to market CompuChess, JS&A entered the market with an almost identical product. Taking advantage of recent technological advances, JS&A apparently "unloaded" the ROM of Data Cash's device, so that in effect the object code was directly transferred to the ROM of JS&A's device.[23]

Data Cash was unable to secure copyright relief against the appropriation of its program, for several reasons. One of major significance here was that JS&A had not made a "copy" of Data Cash's copyrighted source program. A copy of a source program is a source program, according to the trial court, not object code or

---

[19] See *Synercom Technology, Inc. v. University Computing Co.*, 462 F. Supp. 1003 (N.D. Tex. 1978). In the *Synercom* case the court held it no copyright infringement to use the information in a copyrighted instruction manual for a computer program to produce a competing program that would give the same result when presented with the same input data. The input data format, itself, was held uncopyrightable, because the idea and expression had merged and were inseparable.

[20] See, *e.g.*, *id.* at 1013 n. 5.

[21] If a plagiarist took Shakespeare's play *Hamlet* and changed Hamlet's name to John, Ophelia's to Mary, Gertrude's to Jane, Laertes' to Harry, and so on; shifted the action from Denmark to Italy; and translated the words from English to French—we might still recognize striking similarities of detail, such as John's grappling with Harry in Mary's grave, or Jane's drinking the poisoned cup of wine intended for John, see *Heim v. Universal Pictures Co.*, 154 F.2d 480, 488 (2d Cir. 1946); and surely we would still recognize the pattern of the play and thus the plagiarism. See *Nichols v. Universal Pictures Co.*, 45 F.2d 119 (2d Cir. 1930).

But suppose the copyist were free to do all this and also to shuffle the acts, scenes, and lines almost like a deck of cards; to change Hamlet's desire to revenge Claudius' murder of his father to an unwillingness to invite Polonius to a dinner party, to change Ophelia's going mad to her learning to play the flute, and to make Horatio and Laertes Hamlet's grandmother and grandfather. At some point the pattern would become so unrecognizable that it could no longer be said that there was a copy. The software plagiarist can easily do all of this, perhaps with the aid of a computer to break up the apparent pattern still further and more complexly without losing the inner action. Moreover, he has no problem of maintaining at least a minimum of literary merit, so that while changing the play of Hamlet's revenge to that of the inhospitality of John, Doge of Venice, may be a literary disaster, the proof of a program is only whether it works.

[22] *Supra* note 1. The judgment was affirmed on grounds not discussed in text—*viz.*, failure by plaintiff to use a proper copyright notice.

[23] The *Data Cash* court stated that it did not know how the ROM in defendant's product was created. 480 F. Supp. at 1071. It noted, however, that the parties stipulated that defendant's supplier manufactured the ROM from a punched tape that it had received from a Hong Kong company. The parties "speculated" that someone "unloaded" plaintiff's ROM (*i.e.*, by means of a computer decoded the object program stored within the ROM) and then either printed out the program or placed the signals unloaded from plaintiff's ROM into another ROM. Either way, ROMs could then be made that duplicated plaintiff's ROM. *Id.*, n. 14.

a mechanical device, such as a programmed ROM. As the court said, "[T] he ROM is not a 'copy' of the plaintiff's computer program and therefore the [so-called] copying is not actionable."[24] Although the court's ruling was made under the 1909 United States Copyright Act, which preceded the present 1976 Act, the court indicated that for constitutional reasons it would reach the same result under the 1976 Act.[25] Part II.C of this paper treats the same issue under the current United States copyright laws.

C. Effect of 1976 and 1980 Laws on Object Code

Federal legislation was enacted, at the end of 1980, amending the copyright laws concerning software. In the 1976 general revision of the United States copyright law, Congress had found it impossible to secure agreement on the proper scope or application of copyright law to computers. Accordingly, Congress legislated in section 117 of the 1976 Act that the state of the law as to copyrightability of computer material would be left wherever it was on December 31, 1977.[26] At the same time, Congress directed that a Commission report to it with recommendations for copyright legislation as to various computer-related matters.[27] The Commission subsequently recommended legislation of remarkable obscurity,[28] which Congress enacted several years later, in late 1980, as a small part of a large package of patent legislation.[29]

The 1980 law overtly did two things:

(1) It added a definition of "computer program" as a "set of statements or instructions to be used directly or indirectly in a computer in order to bring about a certain result."[30]

(2) It repealed section 117 and replaced it with a section simply stating that it is *not* an infringement for the owner of a copy of a computer program to make copies or adaptations needed in utilizing the program or for archival purposes.[31]

The legislative history of the 1980 law focuses primarily on patent legislation found in the same bill, and says little more concerning software than that the new law defines "computer program" and amends the old law "in regard to copyrights on computer programs."[32]

Some have viewed the 1980 legislation as dramatically broadening copyright protection on software. They regard its declaration that certain acts with respect to programs are *not* copyright infringements as a statement, by implication, that any other generally similar acts *are* infringements. Some are so enthusiastic as to regard the language as a legislative reversal of the *Data Cash* decision's denial of protection to object code. The premise of this reasoning is that the other parts of the 1976 copyright revision law would have already accomplished these results, but for the now deleted section 117 that left the law where it was. According to this view, the removal of section 117 operated like a "discovered check" in chess, unleashing and permitting full operation now of a previously partly veiled or curbed statutory scheme.[33]

Whether the 1980 amendment does any such thing necessarily depends on whether the now fully unleashed 1976 copyright law changed anything as to object code software. First, did the 1976 law make object code copyrightable as such, without reference to source programs or anything else? If not, did it prohibit the unloading, duplication, and marketing of object code as impermissible infringement of any underlying work embodied in a copyrighted source program? Under the old copyright law, these would be different questions with independent answers. Under the new law, it appears that these are merely different formulations of the question of whether object code is a "copy" of a work of authorship.

No decision under the United States copyright laws appears to have analyzed the copyright status of object code, except for the *Data Cash* opinion. It suggests that object code is not a writing and thus was statutorily uncopyrightable under the old law and is constitution-

[24] The court relied on precedents holding that structures embodying designs shown in drawings are not "copies" of the drawings. See, *e.g.*, *DeSilva Construction Corp. v. Herrald*, 213 F. Supp. 184 (N.D. Fla. 1962); *Muller v. Triborough Bridge Authority*, 43 F. Supp. 298 (S.D.N.Y. 1942). The law is different in the U.K. See, *e.g.*, *LB Plastics Ltd. v. Swish Products Ltd.*, [1979] F.S.R. 145 (H.L.).

Not only are structures not "copies" of a writing, under United States law, but they are undoubtedly not "writings" themselves. Congress is authorized to pass copyright laws only to secure to authors the exclusive rights to their writings, *see* note 12 *supra*, and it is highly questionable that the grant of power to protect rights to writings embraces a power to protect non-writings that are related to the protected writings. That does not necessarily mean, however, that non-writings of that sort are not protectable under legislation based on the Commerce Clause or some other part of the United States Constitution.

[25] 480 F. Supp. at 1066-67 n. 4. The Seventh Circuit affirmed on wholly different grounds, refusing to pass on the issue decided below. Data Cash had marketed several thousand copies of the machine (ROM and all), without any copyright notice. It had done so because it mistakenly believed that a ROM could not be unloaded. So extensive a "publication" without proper copyright notice, the court held, worked a forfeiture of any copyright Data Cash had. 628 F.2d at 1041-43. It is possible to regard the decision as a holding, by implication, that if Data Cash had used a proper notice it would have secured copyright protection for the ROM's object code. But that is a highly speculative interpretation of an opinion that does not address the question directly.

[26] 90 Stat. 2541 (1976), formerly codified as 17 U.S.C. § 117. The decision not to legislate as to computers was made about eight years before. See Final Rep. of Nat'l Comm'n on New Technological Uses of Copyrighted Works (hereafter cited as "CONTU Rep.") 4 (1978).

[27] *Id.*

[28] *Id.* at 12.

[29] Pub. L. 96-517 (Dec. 12, 1980), 94 Stat. 3015. The statute primarily concerns government ownership of patents and administrative reexamination of patents.

[30] *Id.*, amending 17 U.S.C. § 101.

[31] *Id.*, amending 17 U.S.C. § 117.

[32] H.R. Rep. No. 96-1307, Part 2, 96th Cong., 2d Sess. 16 (1980). See also, *id.*, Part 1, at 23-24.

[33] See, *e.g.*, CONTU Rep. 12-13.

ally uncopyrightable under either law. [34] We must therefore turn to a textual analysis of the new law, first, and then perhaps to the Constitution, as well.

As already indicated, there are several possible ways in which object code could be protected by the copyright laws:

(1) Object code could be deemed a "copy" of a source program, and thus an infringement of it. This is the most important possible route to protection.

(2) Object code could be copyrightable as such. Copying it would then be an infringement.

(3) An object code counterpart of a copyrighted source program could be deemed a derivative work based on the source program. Preparation of a derivative work is an infringement of the underlying work.

1. Copying Source Programs

Copies. "Copies" are partially defined in the statute as material objects in which a work is fixed for the first or a subsequent time; [35] further, the work must be capable of being "perceived, reproduced, or otherwise communicated" from the copy, either directly or "with the aid of a machine." [36] In one view, this definition compels the conclusion that object code is a copy of the source program: "Because works in computer storage may be repeatedly reproduced, they are fixed and, therefore, are copies." [37]

Intelligibility. A second and opposing view is that the statutory requirement that the work be capable of being "perceived... or *otherwise* communicated" (emphasis supplied) from the alleged copy requires the possibility of perception by a human being—that is, apperception, rather than mere mechanical registration—which is to say, communication and comprehension, at least potentially, of a message that has some intellectual or emotional content. [38] In this view, object code, because it is unintelligible in itself, in not a "copy" of a programmer's expression of his programming idea, just as a cake is not an intelligible copy of the recipe writer's expression and a machine is not an intelligible copy of the draftsman's drawing. Indeed, object code is even less intelligible than a cake or machine. The latter do not communicate an intelligible message, but they are directly perceivable with some senses and convey some impressions or sensations. Object code itself communicates nothing at all, in terms of human reaction or sensibility.

The objection to the copyrightability of object code, in this view, has nothing to do with its being embodied in an unconventional literary medium, such as tape, disk, or ROM. In the first place, source code can be embodied in such media, and at the command PRINT it can immediately be called forth in human-readable and intelligible or comprehensible form (a "copy"). Audiovisual material can also be embodied in such media (*e.g.*, videotape). Moreover, the legislative history broadly indicates that copies can be embodied in any kind of recording medium, so long as the other provisions of law are satisfied. It is proper to conclude, therefore, that object code, when it produces an intelligible display, has the display "fixed" in it as a work, as videotape may fix an opera or motion picture. The problem with object code is not the medium in which it is embodied.

Same Output. The real problem with object programs as "copies" is the absence of a fixed, unvarying intelligible output of a computer directed by object code. The intelligible output is the display or print-out, and that output varies relative to input all the time. Thus, the object code counterpart of a source program differs significantly from a cassette or videotape in which a work such as *Don Giovanni* or *Casablanca* is fixed or embodied. The latter is fixed in a way that always results in the same output: Leporello will always have 1,003 entries for Spain in the Don's catalog, never 999 or 1,004; Bogart will always leave Casablanca at the end to join the Free French Army, and he will never get on the airplane for Lisbon. But the output display caused by a program on a disk or ROM will never be the same twice, and thus causing the same output a second time, unless the two inputs are identical. For a complex program, the probability of the same input's occurring twice is of the same order of magnitude as the probability that a roomful of chimpanzees sitting before typewriters for 100 years will produce a typescript of *Hamlet*, or that a kaleidoscope will be the same twice.

Because the output of object code is input-dependent, object code's output cannot be said to be a copy of any pre-existing work. In a manner of speaking, the input and the object code "co-author" a new work every time. Thus, the possibly intelligible output or display caused by the object code cannot be considered the work embodied in the object code, in the way that *Don*

---

[34] Notes 24 and 25 *supra*.

[35] 17 U.S.C. § 101.

[36] *Id.* The permissibility of using a machine to aid in perception would not seem to be significant. The legislative history indicates that the reference to "aid of a machine" was intended to permit the copyrighting of phonograph records, videotapes, and the like, which permit a listener or viewer to perceive a fixed, human-intelligible message intended to be perceived and appreciated with the assistance of a phonograph, TV, or the like.

[37] CONTU Rep. 22.

[38] *Id.* at 32. The April 27, 1977, Draft Report of the Software Subcommittee, p. 10, recommended that the Commission take the positions that (1) object code has no more "lack of communicative potential" than a book written in Sanscrit or a table of trigonometric functions, which are clearly copyrightable, and (2) if lines must be drawn between copyrightable and uncopyrightable forms of programs, the judiciary should be assigned the task, on a case-by-case basis. The final version of the report states only the second of these positions. CONTU Rep. 22-23.

Giovanni is embodied in a tape cassette. Rather, the embodied work, if anything, must be the program itself, for that is the only other possible work to be found. Object code, however, unlike the intelligible display that it may cause, is not itself intelligible, for a printout of object code is unintelligible even to trained observers.

On balance, the statutory definition of "copy" would seem to make human-intelligibility a *sine qua non* of anything's being a "copy," as that term is used in the Act. Object code is therefore *not* a copy of any other thing, such as source code, that is itself copyrightable.

Reverse Compilation. The counter-argument may be advanced that object code does embody a work (the source program) in a form from which the work can be perceived or otherwise communicated, given the aid of a properly programmed computer. That is, a person having the original compiler program used to compile the object code from the source program, and knowing the other relevant machine parameters, could devise a computer program that will "inverse compile" or "reverse compile" object code and transform it into intelligible source code. Despite some possible slippage in the process, the resulting source code will probably be close enough to the original source program to constitute a "copy" of it. For this reason, it may be said, the original work may be derived in a perceivable and intelligible fixed form (source code) from the object code, at least in some circumstances, and therefore object code is in principle intelligible, not directly but with the aid of a machine to process it.

The argument would not appear to be sound, however. That comprehensible information (the source program) can at times be extracted by a transformation process, from a device not intended to function as a medium for human communication of such information, does not make the device itself an object from which the work can be perceived or otherwise communicated, even in those circumstances when the transformation is feasible. The argument simply proves too much. For example, a work such as a blueprint of a chair or structure could similarly be "reverse-compiled" from the actual chair or structure, by photographic and computer means. But that does not make the chair or structure a "copy" of the work from which the work can be perceived. The law is quite against extending copyright protection to such "copies."[39]

"Used Directly." The 1980 amendments do not change the 1976 Act's definition of "copies." If anything, the 1980 Act would seem merely to reinforce the notion that copies of programs must be in intelligible form for them to qualify as protectable under the copyright laws. As indicated, the 1980 Act overtly did only two things. First, it defined computer programs as a set of statements or instructions to be used "directly or indirectly" in a computer to bring about a result. One might argue that "directly" must be a reference to object code, since a source program is used in a computer "indirectly," in a manner of speaking, because it must be transformed into object code or other machine-readable form to be used in a computer. This is a very slender reed, however, for there is no legislative explanation supporting this interpretation. Moreover, nothing in the 1980 statute states that all computer programs, so defined, are potentially copyrightable. There is no reference in the reports on the 1976 and 1980 copyright laws to the copyrightability of object code, as distinguished from that of "computer programs" as a broad, unspecific category.

Adaptation. The second thing the 1980 amendment did was to delete former section 117 and substitute a new one. The new section declared that it is *not* an infringement for a rightful owner of a program to adapt the program. The report to Congress recommending enactment of section 117 gave two examples of adaptation of programs.[40] Both of them involved source code rather than object code. One example was "conversion of a program from one higher-level language to another to facilitate use"—*e.g.*, conversion from BASIC to FORTRAN.[41] This is clearly a reference to source programs, for only they are written in higher level languages.[42] The other example was "to add features to the program," again a procedure that calls for a human being to revise a program intelligible to him, which necessarily means doing so by dealing with an intelligible copy (*i.e.*, one written in source code, not object code). It therefore remains very doubtful that object code is protectable as a "copy" of the underlying source code, and the 1980 amendments have not affected this conclusion.

2. *Direct Copyrightability*

Object code would also appear not subject to copyright directly as a work of authorship. A work is "created" under the new Act when it is fixed for the first time in a copy.[43] A work is "fixed" when it is embodied in a copy from which it can be perceived or otherwise communicated.[44] Since object code cannot be perceived

[39] See, *e.g.*, *DeSilva Construction Corp. v. Herrald*, 213 F. Supp. 184, 196 (M.D. Fla. 1962); *Muller v. Triborough Bridge Authority*, 43 F. Supp. 298 (S.D.N.Y. 1942); *Jack Adelman, Inc. v. Sonner's & Gordon, Inc.*, 112 F. Supp. 187 (S.D.N.Y. 1934).

[40] CONTU Rep. 13.

[41] See note 20 *supra* and accompanying text.

[42] See text at 5 *supra*.

[43] 17 U.S.C. § 101.

[44] *Id.* The statute is quite repetitious in its references to things being perceived or otherwise communicated. The phrase occurs in 17 U.S.C. § 101 in the definitions of "copies," "fixed," and "phonorecords." It also occurs in 17 U.S.C. § 102, defining the subject matter of copyright. Further, the terms "copies" and "fixed" are used to define various other concepts, as well as each other (circularly).

by or communicated to a human perceiver or communicatee, it is not a "copy" and thus not a work subject to copyright. In addition, object code does not fit within any of the seven classes of copyrightable work set forth in section 102(c).[45]

3. Derivative Work

Object code is not a "derivate work," derived from a source program. A derivative work is a new work of authorship based on, and containing perceptible and recognizable elements of, an earlier work. Examples are a translation of *Hamlet* into French, an edited or modified version of *Hamlet*, a dramatization of one of Boccaccio's tales (such as *All's Well That Ends Well*), or the like. The copyrightable elements (if any) of a derivative work are only those *not* taken from the earlier work, but rather newly created by and original with the author of the derivative work.[46]

These characteristics of a derivative work are inconsistent with according object code that status. First, it is questionable that compiling a source program into object code involves even the small amount of creativity and originality required for a derivative work,[47] since such compilation is routinely done by computer. In any event, there is the same "copy" problem with derivative works as with the other approaches to copyrightability already discussed. Like a source program or any other work of authorship, a derivative work based on a source program must be fixed in a copy, the copy must be perceivable by or otherwise communicated to human beings, and so on. Object code would therefore seem to have no higher status as a derivative work than it does as an original work or as a copy of an original source code work.

4. Utility

The fundamental problem with object code, is, as discussed so far, that it is *not* a medium intended for communication of its content to humans. The other side of the coin is what object code is intended for—operating a machine and doing useful work. A traditional principle of U.S. copyright law, carried forward in the new Act, is that copyright does not protect useful or utilitarian aspects of works. Copyright law traditionally has distinguished between useful and utilitarian articles, on one side, and articles that communicate information between persons, on the other. Chairs, light fixtures, bridges and machines are examples of the former; books, musical scores, maps, and posters, of the latter.

The current U.S. copyright law continues this distinction by making useful articles works of authorship only "insofar as their form but not their mechanical or utilitarian aspects are concerned,"[48] and only if and to the extent that their design incorporates intellectually-apprehended features that can be identified separately from and can exist independently of the article's utilitarian aspects.[49] Similarly, copyright protection is expressly withheld from any "idea, procedure, process, system, method of operation, concept, principle, or discovery, regardless of the form in which it is described, explained, illustrated, or embodied" in a work.[50]

When the primary intended use of an article is utilitarian, in the sense that the article itself performs a mechanical or utilitarian function or it interacts with machines (*e.g.*, computers) to do so, the article would appear to be outside U.S. copyright law. The definition of "copy" reflects this, and so do the other passages of the copyright law, discussed above. In the writer's view, therefore, only when and to the extent that the intended use of an article is as a medium for communication is the article within the copyright law's protection. Object code is a medium for storing data, but not intelligible, communicable data. In no sense is object code a medium of communication, and it is therefore not copyrightable.

This point may need some clarification. Object code in this context means the machine readable counterpart of source code in a source program. It does *not* mean intelligible data (*e.g.*, statements in a natural language such as English or French, or in a human-intelligible language such as BASIC) directly stored in digital form on disk, tape, or ROM. A disk, tape, or ROM used to store such data, and from which the data can be retrieved and directly displayed in human-intelligible form, would be analogous to a book or motion picture film. But a disk or ROM used to store a program in machine-readable form, unintelligible to humans, would be a useful article.

The utility/communication dichotomy reflects the two basic purposes and directions of the U.S. Constitution's provisions concerning intellectual property, and the implementing statutes. Article I, section 8, clause 8 of the U.S. Constitution has two branches, tabulated below:

[45] Section 102(a) states that works of authorship include the following categories: (1) literary works, (2) musical works, (3) dramatic works, (4) pantomimes and choreographic works, (5) pictorial works, (6) motion pictures and other audiovisual works, and (7) sound recordings. There perhaps may be other types of work unlisted in section 102(a), but no suggestion has been made of what they are, and there is no suggestion that object code is such a work.

[46] See *L. Batlin & Son, Inc. v. Snyder*, 536 F.2d 486 (2d Cir. 1976).

[47] See *id.* at 488.

[48] 17 U.S.C. § 101.

[49] *Id.*

[50] 17 U.S.C. § 102(b).

Patent Branch	*Copyright Branch*
To promote progress of useful arts	To promote progress of knowledge
By granting patents to inventors	By granting copyright to authors
To protect discoveries	To protect writings
Utilitarian aspects	Communicative aspects.[51]

The two implementing statutes, the U.S. patent and copyright laws, embody the same dichotomy. Patents protect new and useful advances in machines, processes, and the like, while copyrights protect original works of authorship.[52] Object code, as a practical matter, falls between these two statutory schemes.[53]

To be sure, the question of copyrightability of object code is not free from doubt, and there are no definitive judicial rulings. Accordingly, there is ample room for argument in the field. At the present moment, however, it would be imprudent to expect much more from United States copyright law than prior software decisions have offered. The new law may have covertly accomplished what its enthusiasts now claim, but this will be shown, if at all, only after considerable further litigation of the issue.

In the present state of learning, it appears that the better view is that to be protectable under copyright law, an alleged copy must be intended to communicate some message intelligible to human beings, even if they need a machine to aid in the communication. That an object intended for a utilitarian purpose may be made to disgorge something intelligible to human beings if placed in a proper machine is probably insufficient. By these criteria, object code is not a copy of a work of authorship, and is thus not directly protectable by copyright nor indirectly protectable under the theory that it is an infringement of a copyright based on the source program. Perhaps, Congress will presently enact a less opaque statute, unequivocally declaring that it is a copyright infringement of a copyrighted source program to unload and duplicate an object code counterpart of the source program, but unless and until it does so such conduct will probably continue to escape the reach of the United States copyright laws.

D. Importation of Object Code

Some further potential problems with protecting object code under United States copyright law are suggested by the facts of the *Data Cash* case. These are jurisdictional or territorial. In some cases the copying and the sale of the copy may take place only outside the United States. Thus, a copyist in Hong Kong might unauthorizedly acquire or copy (*e.g.*, photocopy) a source program. The source program may then be compiled in Hong Kong as object code, which is embodied in machines in Hong Kong or in the United States. The machines may then be sold and used in the United States. There would apparently be no copyright infringement actionable under United States law.[54] This highlights a basic problem when United States copyright law is sought to be used to protect software: commercial use of software is not copyright infringement, because use (*e.g.*, execution of a program) is not copying.[55] This is one of the fundamental defects in copyright as a means of protecting software in the United States.[56]

Unfair Importation. It should be noted, however, that United States law may afford some protection against imports in situations like that of the *Data Cash* case, under Section 337 of the Tariff Act.[57] That law prohibits

[51] Article I, § 8, cl. 8 provides that Congress shall have power "To promote the Progress of Science and useful Arts by securing, for limited Times to Authors and Inventors, the exclusive Right to their respective Writings and Discoveries." This provision follows 18th century parallel structure syntax, which provides these two branches: (1) "To promote the Progress of Science... by securing to Authors... the exclusive Right to their Writings," and (2) "To promote the Progress of... useful Arts by securing to... Inventors... the exclusive Right to their... Discoveries." The term "Science" (*scientia*) is equated to general knowledge. See generally *Graham v. John Deere Co.*, 383 U.S. 1, 5 (1966).

[52] The distinction is forcefully expounded in *Baker v. Selden*, 101 U.S. 99 (1879). In that decision, the Supreme Court denied copyright protection to the new accounting system explained by the plaintiff in his book. In so ruling, the court explained that copyright law deals with works "conveying information... and containing detailed explanations" while patent law deals with the use of information conveyed by books or other means of communication. The Court noted that some advances, such as perhaps that made by plaintiff, could not be protected under either system.

Patent rulings make the same distinction. See, *e.g.*, *Kaz Mfg. Co. v. Chesebrough-Ponds, Inc.*, 317 F.2d 679 (2d Cir. 1963) (collecting authorities on use of inventions only for communication, rather than practice of invention itself, as not being patent infringement).

[53] It may be original, but as a non-writing it is not copyrightable. Generally, even though it is original, it is usually not novel enough to meet the patent law's inventiveness requirements. Object code is thus like the subject matter of a petty patent or utility model, which is useful but not inventive. Present United States law simply does not protect that kind of subject matter.

[54] Importation into the United States of a "copy" of a work acquired outside the United States is an infringement, however, unless the importer imported only a single copy for his private use and not for distribution. 17 U.S.C. § 602(a)(2). It could be argued that "private use" does not include use for commercial or business purposes, but this is a questionable interpretation. The more important problem, however, is whether a "copy" is involved.

[55] Under the principle established in *Baker v. Selden*, 101 U.S. 99 (1879), commercial use of concrete ideas set forth in literary works is not a copyright violation. Thus, under United States law, it is not copyright infringement to bake a cake in accordance with a recipe printed in a cookbook, nor to knit and sell a sweater described in a knitting instruction book.

Execution of the program is not in itself the making of a copy any more than baking the cake is the making of a copy of the recipe. See, *e.g.*, *DeSilva Construction Corp. v. Herrald*, 213 F. Supp. 184 (N.D. Fla. 1962); *Muller v. Triborough Bridge Authority*, 43 F. Supp. 298 (S.D.N.Y. 1942). But see *L.B. Plastics Ltd. v. Swish Products Ltd.*, [1979] F.S.R. 145 (H.L.).

[56] Under United States patent law, in contrast, unauthorized use of the subject matter of a patented invention is infringement, as are manufacture and sale.

[57] 19 U.S.C. § 1337.

"unfair" methods of competition in import commerce and directs the United States International Trade Commission to prohibit such conduct when it threatens substantial injury to an economically and efficiently operated domestic industry. Infringement of a patent, trademark, or copyright can be prevented under this statute. Moreover, Section 337's concept of "unfair" is not limited to conduct prohibited by the patent, copyright, and analogous laws.[58] Thus, importation of ROMs unloaded from a domestic complainant's ROMs might be an unfair method of competition, on the theory that such conduct violates the "spirit" of the United States copyright laws.[59] Accordingly, even though no copyright relief against appropriation of object code is available domestically,[60] it may be possible for software proprietors to prevent importation of devices embodying "unloaded" object code.

* * *

To recapitulate, United States copyright law affords meaningful protection to software only when it is in a form akin to a literary work, usually only a source program. Even this protection is limited by the ease with which a source program can be redone as a different source program that may not readily be shown to be derived from the first one. Accordingly, unless the software proprietor can show true copying of a source program (*i.e.*, not merely utilizing its concept or using the program, but actually making another source program identical or substantially similar to the copyrighted source program), relief against major commercial uses of software is probably not available under United States copyright law.

[58] *Certain Apparatus for the Continuous Production of Copper Rod*, I.T.C. Dkt. No. 337-TA-84, Opinion of Nov. 6, 1980.

[59] This question was raised recently in *Certain Coin-Operated Audiovisual Games and Components Thereof*, I.T.C. Dkt. No. 337-TA-87. The case involved importation of coin-operated, space-invader, video games, apparently containing ROMs unloaded from the ROMs used by the complaining party. The ROMs produced two kinds of visual display. One was a constant sequence of visual images. The other involved a different sequence of images in each game played, because the ROM cooperated with a microprocessor to produce a display dependent on player input. It was suggested to the Commission (by the writer): (1) that the former display was a copyright infringement; (2) that while the latter was not; and (3) that the latter nonetheless violated the "spirit" of the copyright laws and should therefore be treated as an unfair method of competition. The Commission did not reach the second and third questions, however, because it found a copyright law violation in the copying of the former (constant sequence) display material. Opinion of June 25, 1981.

[60] It may be argued that state unfair competition law or section 5 of the Federal Trade Commission Act, 15 U.S.C. § 45 (prohibiting unfair methods of competition and unfair acts and practices), can provide domestic relief against such "unfair" acts. As for relief under state law, see note 1 *supra*. As for relief under the FTC Act, there is some slight authority to support this view. See, *e.g.*, *FTC v. Orient Music Roll Co.*, 2 F.T.C. 176 (1919). But it is doubtful that the agency would have the inclination to provide effective such protection in most instances. Moreover, it may well lack authority to do so, because it is required to act only to protect the general public rather than to redress private wrongs. *FTC v. Klesner*, 280 U.S. 19 (1929).

III. Patents

Until recently, no form of protection for software *per se* or when used with machine systems or industrial processes seemed available under United States patent law. A 1981 decision of the United States Supreme Court, however, may indicate a change in the law.[61]

Benson. The Supreme Court first considered the patentability of computer algorithms in the *Benson* case in 1972.[62] Like all other Supreme Court software decisions, the case involved an appeal from an administrative denial of a patent grant. The applicant sought a patent on a method of converting numerical representations coded in one format into those coded in another format—essentially, an algorithm for conversion from decimal notation to binary notation.[63] Although the applicant (Bell Telephone) asserted that the method was intended to be used only on signals in telephone switchboards, the claims of the patent were not so limited and would have covered any use of the method. In upholding the Patent Office's refusal to allow a patent, the Court noted that the patent if granted would preempt all use of the algorithm for any end use, whether in telephony or otherwise. To the Court, this would be equivalent to patenting the algorithm and thus an idea, which is clearly not permitted under the United States patent laws.

Flook. Subsequently, in the *Flook* case, the Court seemed even more emphatically to rule out patents on software.[64] A patent was sought on a method for "updating alarm limits" during catalytic cracking or conversion of hydrocarbons. A supposedly novel algorithm was used, in conjunction with monitored variables (such as reaction temperature) to determine whether the present value of the variable had changed so substantially from previous values as to suggest that a runaway reaction was about to occur.[65] In *Flook*, the end use was limited to catalytic cracking—unlike the unlimited scope of the *Benson* claims. Yet, the Court still found the claim unpatentable, because the algorithm (which the Court said must be assumed upatentable *per se*) had not been applied in Flook's method in a novel, inventive manner. After *Flook*, it appeared that software was

[61] *Diamond v. Diehr*, 101 S.Ct. 1048 (1981).

[62] *Gottschalk v. Benson*, 409 U.S. 63 (1972).

[63] The input signals to the computer were in binary-coded decimal ("BCD") format, a series of four pulses (of "0" or "1" magnitude) per decimal digit. The computer converted the signal to a single series of pulses (0's and 1's, again) representing the entire number in pure binary format. Thus, ordinary decimal 61 is 0110 0001 in BCD and 111101 in pure binary. The computer algorithm was a method for making this conversion. The outcome of the case may have been colored by the fact that the algorithm, was, in substance, multiplication of the 10's digit by 10 and adding the units digit (*e.g.*, 61 = (6x10)+1), performed in accordance with the rules of binary arithmetic.

[64] *Parker v. Flook*, 437 U.S. 584 (1978).

[65] As in *Benson*, there was some question about the novelty of the algorithm. It involved measurement of the current value of the variable, comparing that value with a weighted average of past values, using a predetermined weighting or smoothing factor, and taking action only if the difference exceeded a predetermined safety factor.

either always unpatentable, or patentable only when applied in a novel, unexpected, and unobvious way; and if software were patentable, the inventive step had to be displayed in the mode of the software's application, not in the software itself.

Diehr. In March 1981, however, the Supreme Court abruptly reversed field in the *Diehr* case, on what seem to be facts no different from those involved in *Flook.* This time, the method was a process for curing rubber, in which the molding press was opened when elapsed time and temperature data satisfied an equation (the "Arrhenius equation") relating these variables and other parameters. Although the Arrhenius equation was old,[66] as was the molding press itself and its use in curing rubber, apparently the combination of all the elements was new and produced unusually superior results. The Court allowed the patent and distinguished *Flook* on the ground that there the patent application provided nothing but a mere formula, while the instant application disclosed a process for curing rubber,[67] which incorporated a more efficient solution to the problem of curing rubber for the correct length of time. Although it appears that each of the steps in the claimed rubber-curing method was old, or obvious when considered in isolation,[68] nonetheless the Court appears to have regarded their combination to be inventive, apparently because of the superior results achieved.

The present state of United States law is unclear, but it still appears that software *per se* is unpatentable. When software is combined with something else, however, such as a number of steps in an industrial process, or a machine system for carrying out such an industrial process, the totality may be patentable.[69]

Breadth. Patent protection of such software is broader than copyright protection would be. Anyone who makes, uses, or sells a patented machine system is an infringer, and therefore may be enjoined and required to pay damages (which may be multiplied if the infringement is deliberate). Making or selling a non-staple device particularly adapted to use in or to be part of an infringing machine system is also infringement.[70] Thus, manufacture and sale of a ROM in which Diehr's program has been loaded is arguably an infringement. Certainly, execution of the claimed program in the course of carrying out the rubber-curing method of the Diehr patent would be an infringement. Finally, the device or process that infringes the patent need not be "substantially similar" in form or appearance to the original; infringement occurs if an equivalent device or process is used, *i.e.*, one that does the same thing in the same way. It is thus seen that validly patented software systems are far more secure from infringement than copyrighted software is.

Although no Supreme Court decision has involved programs, and all have involved algorithms, it would seem that, if an algorithm in combination with or embedded in the context of a process or machine is potentially patentable, so too is a program.[71] Potentially broad protection of software, therefore, may be available under the patent system. As a practical matter,

[66] The Court said: "The equation is named after its discoverer Svante Arrhenius and has long been used to calculate the cure time in rubber molding presses." *Diamond v. Diehr,* note 61 *supra,* 101 S.Ct. at 1052, n.2.

[67] It is unclear whether one result of the *Diehr* decision will be to allow patents on "applications" software but not on "systems" software. Until now, this was not a legally significant distinction. "Applications" software is used to automate particular business tasks, such as inventory control, accounts receivable, and payroll, and it usually is marketed in standard packages for that task. "Systems" software is used to operate computers, generally, and makes it easier or more economical for the computer user to write software and otherwise use the computer for data processing. It is estimated that although sales of applications and system software packages are now about equal, within five years the overwhelming majority (perhaps 80 percent) of software package sales will be of applications software. *Business Week,* Sept. 1, 1980, at 55.

The software involved in *Diehr* was analogous to applications software, while that involved in *Benson* was analogous to systems software. One of the points stressed in *Benson* and other software patent decisions was that allowing the patent sought would preempt all possible end use of the algorithm. The algorithm involved in the *Flook* case had many potential end uses, and the patent applicant sought to avoid the "preempt all end use" argument by arbitrarily carving out one end use (hydrocarbon cracking) and waiving or abandoning to the public all other end uses. The Court considered this a transparent subterfuge. In the *Diehr* case, the apparatus and other limitations necessarily confined the scope of the patent to one application (molding and curing cast rubber articles), and the Court did not condemn the patent applicant as seeking to preempt all possible uses of the algorithm.

Generally, applications software, if patentable, is likely to be confined to the particular application in question, *e.g.*, inventory control. Systems software, however, is likely to be limited only to use in digital computers. Accordingly, if preemption of all possible end uses is still a valid objection to a software patent, systems software may be more vulnerable to the objection than applications software.

[68] It was unclear whether frequent measurement of the mold temperature (every few seconds) was new or obvious. There was also a disagreement over whether the particular placement of the temperature-measuring device was new or obvious.

[69] The *caveat* is important that software can be patented as part of a machine system or industrial process only when it lends itself to be claimed in a way pointing away from the software aspect of the invention. For example, it is probably necessary for the claims draftsman to include steps around the algorithm, stressing particular methods of monitoring input variables with particular devices, even if these methods and devices are not novel or are obvious. The apparatus controlled by the computer and software probably must be described, and the things done with the output signals from the computer probably must be specified in mechanical terms (*e.g.*, open the mold and remove the rubber tire) rather than conceptual ones (*e.g.*, the output signal is routed to the mold-control device). Otherwise, the courts are likely to conclude that a mere mathematical calculation is being claimed.

It is still unclear what novelty and obviousness standards will be applied to such claimed inventions. WIPO, *Model Provisions* at 5 suggests that "even if patent protection were generally available, it would probably cover only a minute proportion of computer programs since it is considered that only in a few cases (perhaps 1%) would a program have sufficient inventiveness to satisfy the requirements of patent law...."

[70] See 35 U.S.C. § 271(b)-(d); *Dawson Chem. Co. v. Rohm & Haas Co.,* 448 U.S. 176 (1980). The device should not be capable of substantial utility for other, noninfringing purposes.

[71] It may be, however, that there is little point in trying to patent a program if one can patent the algorithm (in the context of a process or machine), for the claim to the algorithm would seem to cover any program derived from it.

however, the delays and difficulties of securing a patent may be too great to justify the effort for most software, particularly if the commercial life of the software is relatively short compared to the several-year delay in securing issuance of a patent and thus protection. Moreover, uncertainty over whether the patent will be held valid and thus enforceable in a subsequent infringement action further limits the attractiveness of securing a patent on software that may be claimable under a patent.

IV. Contract and Trade Secret

By far the most widely used protection for software in the United States is a combination of contract and trade secret protection. It is impractical to speak of these two bodies of law separately, in the software context, because of the interaction of the two. In theory they can be separated, and some cases have involved only one of them. But trade secret rights are often the result of contracts, and contractual restrictions to protect software from misappropriation are often enforceable only because the restrictions are justifiable as necessary to maintain trade secret status.

A. Advantages of Contract—Trade Secret Protection

Explicit contractual arrangements with customers or licensees provide the best means of protecting software rights in the United States, at present, because of the limited value or availability of copyright or patent protection. Contracts are flexible and readily designed to meet the needs of the parties or situation. By the same token, of course, contracts have no standard pattern of customary rights and obligations, so that each transaction may be open to relatively free floating and involved negotiation. Moreover, contract and trade secret law are each matters of state, rather than federal, law. Consequently, they vary widely from state to state, and there is considerable unpredictability of result when multi-state commercial activity is involved. Nevertheless, on balance, contract and trade secret law remain the preferred forms of software protection in the United States.

Generally speaking, a duty not to use a trade secret of another arises under United States law by express contract or by status (*i.e.*, contract or duty implied by law by reason of the relation of the parties). The former is of principal importance in licensing software; the latter is of principal importance in the case of ex-employees and business associates (such as suppliers, subcontractors, and potential partners or joint-venturers). Contractual restraints may be designed to prevent employees or customers from appropriating software or disclosing it to others, but it is unlikely that courts will specifically enforce very substantial agreed-upon restraints unless the software is maintained as a trade secret.

Data General. The effectiveness of contract and trade secret law, in combination, to protect software is illustrated by a decision in a field very closely related to software—logic design for computers. In the *Data General* case,[72] the plaintiff had sold its minicomputers to customers and furnished them drawings for maintenance purposes; the drawings disclosed the inner workings (logic design) of the computer. Plaintiff placed a proprietary notice on all copies it made of the drawings, and plaintiff uniformly secured agreements from all parties to whom it furnished the drawings, to the effect that they would not copy the drawings or use them as the basis for manufacture or sale of minicomputers. Defendant obtained a set of the drawings, from which it developed the design for a competitive minicomputer that was similar in operation to plaintiff's minicomputer and was capable of using any software designed for use on plaintiff's minicomputer.

The state court held that plaintiff's dissemination of the drawings to its customers, under a secrecy agreement and with a proprietary legend, was not a general publication. A general publication would have placed the drawings and the information they contained in the public domain under copyright law. Instead, the dissemination was held to be a limited publication, which maintained the information as trade secrets. The information was therefore protectable against the defendant, who had used the drawings in violation of both the proprietary legend and an express secrecy agreement. The court then held that the circumstances warranted a permanent injunction against defendant and an award of damages.

Although the *Data General* case involved the logic design information in drawings of computer hardware, clearly the same principles would apply to computer programs furnished to a plaintiff's customers under similar contractual and labeling restrictions. In all probability, any defendant that deliberately copied a printout of a program and used it in defiance of a similar uniformly-imposed contractual undertaking and proprietary legend would be held liable in the same way as was the defendant in the *Data General* case.

A number of recent decisions point to this conclusion. For example, in the *Com-Share* case,[73] plaintiff secured an injunction preventing its former licensee from transferring to a successor certain of the software that plaintiff had created and then licensed to defendant under a written agreement prohibiting disclosure or sale. The court specifically recognized software as a type of intellectual property eligible for trade secret protection, referring to the software involved in the case as "unique property that constituted trade secrets of [Com-Share] and were supplied to Computer Complex in confidence

[72] *Data General Corp. v. Digital Computer Controls, Inc.*, 357 A.2d 105 (Del. Ch. 1975).

[73] *Com-Share Inc. v. Computer Complex, Inc.*, 338 F. Supp. 1229 (E.D. Mich. 1971), *aff'd*, 458 F.2d 1341 (6th Cir. 1973).

under [and subject to] restraints against sale, lease, or disclosure set forth" in the parties' contract.[74]

B. Limits of Protection

It is thus seen that software is eligible for contractual and trade secret protection, when proper steps are taken. As a general rule, contracts against unauthorized use or disclosure of proprietary software will be upheld, within limits, and injunctive relief and damages will be awarded against those who breach or induce the breach of such agreements. There are definite limits to the utility of contract and trade secret law, however, as a means of protecting software. A software proprietor's principal concern, in this context, is that his software not be exploited commercially without payment to him. He wants his customers to pay him for using his software and not to disclose it to others or continue to use it once the contract is over. He wants his employees and business associates not to terminate business relations with him and then proceed to exploit his software. These concerns of the software proprietor may collide with other social or policy considerations when he tries to enforce contractual provisions designed to safeguard them. Broad agreements not to engage in the business that the software concerns, after termination of a license, may be desirable to overcome the difficulties an ex-licensor may have in policing the activities of an ex-licensee. Yet, the anticompetitive effects of such restrictions may make them run afoul of federal and state laws against undue restraints of trade.[75] Similar problems may attend post-employment restrictions on former employees.[76]

The general rule in the United States is that such contractual restrictions will be permitted (be enforced by the courts and not give rise to liability to those adversely affected by the restrictions) when they are reasonably adapted to protect the legitimate interest of the software proprietor in maintaining the commercial value of his software. Thus the restraint should not be substantially greater in scope and duration than is reasonably necessary to protect the proprietor from misappropriation of his software. This standard is often designated "the rule of reason" or "the doctrine of necessary and ancillary restraints."

This test suggests the impropriety of certain types of restriction. Under this test, for example, a general prohibition against subsequent competitive activity in the field will probably be deemed excessive, and thus unreasonable, if a more carefully tailored restraint would accomplish the legitimate purposes of the software proprietor.[77] Similarly, a prohibition against any subsequent use of the software proprietor's material, even that which legitimately passes into the public domain, is likely to be deemed unreasonable. The reason is that ordinarily the software licensor or proprietor will not have a legally protectable interest in preventing such use,[78] and pro-competition policies favor widespread commercial use of such public domain material.

On the other hand, the courts will probably uphold a prohibition against any post-license (or post-employment) use of software by the licensee (or ex-employee) as long as the software is maintained in secrecy. (In some states, this may mean merely that the software is not generally known in the trade.)[79] The courts will probably also uphold similar restrictions against publication or disclosure.

---

[74] 338 F. Supp. at 1238.

[75] See, *e.g.*, *Technicon Medical Information Systems Corp. v. Green Bay Packaging, Inc.*, 480 F. Supp. 124 (E.D. Wis. 1979). Plaintiff sought relief against misappropriation of computer software. Defendant counterclaimed under the federal and state antitrust laws, alleging, among other things, that plaintiff's contractual restrictions on the dissemination and use of allegedly secret software illegally restrained trade.

[76] See, *e.g.*, *Cyborg Systems, Inc. v. Management Science America, Inc.*, [1978-1] Trade Cases 61,927 (N.D. Ill. 1978); *Management Science America, Inc. v. Cyborg Systems, Inc.*, [1977-1] Trade Cases 61,472 (N.D. Ill. 1977). In these cases, Management Science sued Cyborg for allegedly stealing Management Science computer software payroll system and competitively marketing it. Cyborg counter-sued Management Science, alleging that it (1) imposed improper restrictive convenants on key Cyborg employees while they had formerly been employees of Management Science, and (2) defamed Cyborg and the ex-employees by bringing a sham trade secret suit for the alleged theft of secret software.

[77] *Restatement, Contracts* § 515(a) (1932). This principle is illustrated by the decision in *Electronic Data Systems Corp. v. Kinder*, 360 F. Supp. 1044, 1048-50 (N.D. Texas), *aff'd*, 497 F.2d 222, 224 (5th Cir. 1974). Plaintiff EDS developed unique software for processing Medicare claims, on which defendant Kinder worked. When he quit to go to work for a competitor, EDS sued to enforce a non-competition covenant, even though Kinder's work at the competitor involved only public domain systems and not EDS' unique software system. The trial court denied relief, because EDS' non-competition agreement was "unreasonable or not necessary for the protection of EDS' business." The court of appeals affirmed, stating: "To restrict him from such employment is not only unnecessary for the protection of the business of EDS, but also is an unwarranted limitation on Kinder's freedom to work as a data processor."

[78] An exception may exist, however, for material licensed to one or only a few persons who thereby gain a head start over the rest of the field. Here, it may be contemplated that the material may fall into the public domain as a result of commercial use, and the agreed-upon consideration may be determined with that in mind. In such circumstances, prohibition of use unless royalties are paid would probably be upheld. See generally *Aronson v. Quick Point Pencil Co.*, 440 U.S. 257 (1979).

[79] This highlights one of the problems, however, in relying on trade secret law to protect software. If a software package is licensed widely enough, it may be deemed no longer to be a trade secret. But see *Management Service America, Inc. v. Cyborg Systems, Inc.*, 6 Computer L. Serv. Rep. 921 (N.D. Ill. 1978) (secrecy provisions of licenses of software possibly effective despite 600 licensees, summary judgment denied).

Moreover, security measures to preserve secrecy must always be kept in force. This may be quite burdensome. In the *Com-Share* case, note 73 *supra*, the software proprietor placed confidentiality notices on each page, used an elaborate password system for access, and kept copies under lock and key. Similar security measures (guards, TV cameras, sensors, computer-controlled access systems) were involved in protecting the software and other computer-related trade secrets misappropriated in *Telex Corp. v. IBM Corp.*, 367 F. Supp. 258, 330-31 (N.D. Okla. 1973) (cost of security measures about $3,000,000), *rev'd on other grounds*, 510 F.2d 894, 932-33 (10th Cir.), *cert. dism'd*, 423 U.S. 802 (1975).

A contract, unlike copyrights and patents, can extend to object programs, algorithms, or anything else. Accordingly, a software proprietor may impose a contractual duty on licensees to whom the proprietor temporarily lends a ROM not to "unload" the ROM[80] or knowingly permit it to be unloaded. In appropriate circumstances, the software proprietor would have an action for tortious interference with contractual relations, against a third party who unloaded a ROM subject to such protection. By the same token, the severe problem with copyright that use is not copying does not apply to a contract.[81] It is unclear, however, whether the same restraint can be enforced against the purchaser of a product in which the ROM is a part.[82] A few years ago the practice would have been clearly illegal, but a recent relaxation of prohibitions against post-sale restraints, when imposed by manufacturers on retailers,[83] may have changed the legal climate for such limitations on customer activity, at least when the number of customers is so small that it can be maintained that there has not been a general release of the ROM-bearing product into commerce.

Nonetheless, policing and enforcement problems may be acute. Thus a software proprietor may contract with a licensee that the latter shall not permit others to copy or use a program. Suppose, however, that the licensee allows the software to "escape," whether carelessly or without fault. It is doubtful that the software proprietor will be able to prove that a remote third party *wrongfully* acquired the proprietor's software, and without such wrong it is unlikely that the proprietor can secure any relief against the third party. First, as far as the law of contract and tortious interference with contract are concerned, unless the third party can be shown to have had knowledge of the license restriction and to have been responsible for its breach, he probably has no liability to the software proprietor.

Second, as far as trade secret law is concerned, it is questionable that a third party who acquires a trade secret (*e.g.*, a mass-marketed software package), in circumstances that the software proprietor cannot prove involved wrongdoing, can be held liable in any way. For the third party's past conduct, the trade secret proprietor most likely cannot obtain such typical trade secret relief as damages, or an accounting for profits due to allegedly unjust enrichment, or on a constructive trust theory. As to future conduct, perhaps an injunction against further use might lie but, if the third party's reliance has led to substantial investment or there are other intervening-rights considerations, it is doubtful that any prospective trade secret relief can be had. Probably, the software proprietor cannot even secure imposition of a royalty requirement.

Moreover, it is unlikely that contract, trade secret, or tort remedies are available to protect a software proprietor's consumer products (or other widely distributed products) embodying object programs. Once a product is widely sold on the open market, the general rule in the United States is that others may "reverse engineer" it,[84] regardless of proprietary notices and contracts between the manufacturer and his retailers.[85]

"Property Right" Problem. The problem with reliance on contract is essentially that software contractual arrangements are unlikely to carry over to third parties, because the courts perceive and recognize no statutory or common law "property right" in software. To be sure, the circularity of property rights and availability of legal remedies is familiar.[86] Nonetheless, because patents and copyrights are statutory property rights, the statutes and case law accord their proprietors a substantial range of remedies against their unauthorized use ("misappropriation"), which are otherwise not available. Appropriation of software is likely to be held "misappropriation," and thus a ground for legal remedies, only when the courts conclude that the defendant has committed a clear breach of contractual or trade secret law duty. Otherwise, remedies will remain unavailable until software legislation comes into being imposing a more extensive range of statutory rights and duties corresponding to those of patent and copyright law.

[To be continued]

[80] See note 23 *supra*. An agreement has been upheld that prohibited a licensee from disassembling a machine or tool, allegedly to protect the secrecy of the inner workings of the tool. *K & G Tool & Service Co. v. G & G Fishing Tool Service*, 158 Tex. 594, 314 S.W.2d 782, *cert. denied*, 358 U.S. 898 (1958). *Accord, National Rejectors, Inc. v. Trieman*, 409 S.W.2d 1, 23 (Mo. 1966) (sealed unit in leased machine).

[81] See text at note 56 *supra*.

[82] See *Merchant Suppliers Paper Co. v. Photo-Marker Corp.*, 29 A.D.2d 94, 285 N.Y.S.2d 932 (1967) (manufacturer cannot by contract prevent inspection of machine sold to others, as a means of preventing copying). The court also relied on federal antitrust law to disapprove such restrictions, as well as on the principle of the *Compco* and *Stiffel* cases, note 1 *supra*.

[83] See *Continental T.V., Inc. v. GTE Sylvania, Inc.*, 433 U.S. 36 (1977).

[84] 1 Milgrim, *Trade Secrets* § 2.05[2] n. 8. See generally *Compco* and *Stiffel* cases, note 1 *supra*.

[85] Note 82 *supra*.

[86] See, *e.g.*, *United States v. Willow River Power Co.*, 324 U.S. 499, 502-03 (1945): "[N]ot all economic interests are 'property rights'; only those economic advantages are 'rights' which have the law back of them, and only when they are so recognized may courts compel others to forbear from interfering with them or to compensate for their invasion.... We cannot start the process of decision by calling such a claim as we have here a 'property right'; whether it is a property right is really the question to be answered. Such economic uses are rights only when they are legally protected interests."

Addendum

Since the author prepared the foregoing material, he has had the benefit of critique and comments, which he feels appropriate to include in the paper.

1. The terminology for source code used here may not coincide with the usage of others. Many in the field designate both assembly code and high-level language (*e.g.*, BASIC) as types of source code. When the term "source code" is used in the introductory part of the paper, it means source code written in a high-level language. Moreover, some persons question whether conversion from one high-level language to another is a straightforward and simple conversion. It is suggested, for example, that translation of a program from BASIC to APL is far more difficult than translation from one natural language to another (such as English to French).

2. The distinction between object code when physically embodied in a ROM, disc, or tape, and object code as a written "printout" or "dump"—a printed sequence of 1's and 0's—is left unclear. Although the author believes that neither form of object code is copyrightable under present United States law, clearly the printout has a greater probability of being copyrightable than the physical form.

3. A substantial number of persons claim to be able to read and rewrite object code (printout form), despite the assertions in opinions of courts to the contrary. This raises the difficult question of how many people must be able to read an alleged "writing" for it to qualify as a writing. In *White-Smith Music Pub. Co. v. Apollo Co.*, 209 U.S. 1 (1908) (a decision now under a cloud), the Court considered music recorded on piano rolls to be too unintelligible to be a "writing" and thus an infringing "copy" of copyrighted sheet music. It so ruled despite the apparent fact that a few persons highly skilled in making piano rolls could read them as other persons read ordinary musical notations. The Court dismissed this fact as pertinent for two reasons. First, the piano rolls were not intended to be read. *Id.* at 18. Second, the argument, if accepted, could be extended to the pattern of metal posts on the cylinders of music boxes and other mechanical devices, which were clearly outside the intended scope of the copyright laws as they then stood. In modern terms, one might refer to "reading" ROMs under a microscope or "reading" electronic circuits on printed circuit boards or chips.

On the other hand, in *Reiss v. National Quotation Bureau*, 276 F. 717 (S.D.N.Y. 1921), a distinguished jurist held that the copyright law would protect a code book made up of 5,000 five-letter "words" of no ordinary meaning at all. Users were to assign their own meanings to these "words" and then use them to economize on cable message costs. This result would seem on a par with treating object code (printout form) as a writing.

It is impossible to determine sensibly how wide a potential audience of readers must exist for a candidate for the status of a "writing" to be accepted. Perhaps, the human-intelligibility argument is a distraction. It may be better to think primarily in terms of intended use—for communication of a message or for operating a machine. Copyright law is intended to protect against copying of expressions of ideas, and not to protect against the use of ideas. ROMs are intended for use in operating machines, and are properly considered unprotectable utilitarian objects. The case as to object code printouts is more equivocal.

4. A recent trial court decision, *Tandy Corp. v. Personal Micro Computers, Inc.*, 524 F. Supp. 171 (N.D. Cal. 1981), rejects the *Data Cash* decision. It takes the position that object code in a ROM is protectable under the copyright laws against competitive unloading and duplication. The analysis is rather flimsy, and the court seems to assume that perception by a computer or communication to a computer is covered by the statute, for it speaks of the computer as reading, understanding, and acting on the program coded into the ROM.

5. Several recent decisions, not yet officially reported, sustain the copyrightability of output displays of video game machines. The decisions do not protect the computer programs loaded into the machines' ROMs, but protect the displays as "audiovisual works." It does not appear, however, that the reasoning of these decisions can be extended beyond the video game field, such as to data processing. The result of the command LIST may be copyrightable, but the result of the command RUN should not.

News from Industrial Property Offices

CANADA

Activities of the Bureau of Corporate Affairs in 1979-1980*

The Bureau is concerned with much of the general legal framework governing the orderly conduct of business under federal jurisdiction and with encouraging innovation by granting temporary legal monopolies to originators of inventive and creative works.

The Bureau of Corporate Affairs and the Bureau of Intellectual Property were amalgamated in October 1979, the combined bureaux being known as the Bureau of Corporate Affairs. This is a return to the original organization at the time the department was created in 1967. The amalgamation was effected because the legislative revision program in bankruptcy and insolvency, incorporation and securities market law had been virtually completed and separate bureaux were no longer necessary. As a result of the amalgamation, the research functions of the separate bureaux were also combined. The Technical Advisory Services Branch, part of the Bureau of Intellectual Property, was abolished on March 31, 1980, and branch operations were phased out or reallocated over the last six months of the year.

The Bureau of Corporate Affairs now consists of Bankruptcy, Corporations, the Patent Office, the Trade Marks Office, the Copyright and Industrial Design Office, and Research and International Affairs.

Patents

The Patent Office administers the Patents Act and Patent Rules, which grant patents for new and often unusual inventions. Examination of patent applications involves a search for novelty, a decision regarding patentability and verification of compliance with procedural requirements.

The inventor or owner of a patent has the right to exclude others from making, using or selling the invention in Canada for 17 years from the date of the grant.

The Office publishes the *Patent Office Record*, a weekly magazine providing details of all Canadian patents granted during the previous week. It also maintains a search room and library where the public may obtain information concerning Canadian and foreign patents.

The Office this year dealt with an average of 453 requests daily. The number of search enquiries from patent agents and the public totalled 3,525, up from 3,513 enquiries last year.

The complete Patent Office file of 1,074,500 issued patents is organized in technological classes which are constantly reviewed, revised or extended as different technologies emerge or as new combinations of known technologies are developed. By March 31, 1980, the file consisted of 340 main classes of technology, which were then condensed into 34,600 subclasses. During the year, 11 classes consisting of 842 subclasses were completely revised; 579 new subclasses were established and 254 old subclasses were abolished in the partial revision of existing classes.

The following table summarizes the transactions of the Patent Office during the past two years:

	1978-79	1979-80
Caveats registered	151	179
Applications for patents	24,312	24,347
Applications restored under Section 75 after failure to pay final fee	192	106
Applications reinstated under Section 32 after abandonment for failure to reply to an official action	181	129
Examiners' Reports	29,055	28,431
Applications allowed	25,133	24,354
Applications forfeited	728	407
Patents issued (for 17 years)	22,772	22,872
Patents reissued	24	22
Patents issued under Public Servants Inventions Act	56	75
Assignments recorded	22,621	21,993
Petitions for a compulsory licence under Sections 41 and 67	13	33
Duly Registered Patent Agents	1,731	1,645
Canadian Resident Agents	—	286
Canadian Firms	—	68
Non-resident	—	1,291

Patent Appeal Board

The Patent Appeal Board reviews, on request to the Commissioner of Patents, final rejections of applications for the grant of patents and for the registration of industrial designs. The Board may, if requested, hold formal hearings as part of the review procedure. Its findings and recommendations are submitted to the Commissioner for approval.

The following table summarizes the Board's activities:

*Excerpted from the Annual Report, Consumer and Commercial Affairs, Canada, for the year ended March 31, 1980.

	1978-79	1979-80
Patents		
Rejections referred to Board	167	185
Rejections affirmed	74	84
Rejections reversed	25	45
Otherwise disposed of	78	29
Hearings held	51	70
Pending	162	200
Decisions published in the Patent Office Record		
full	26	25
part	11	14
Industrial Designs		
Rejections affirmed	6	4
Rejections reversed	0	0
Hearings held	3	2

Compulsory Licences

The Commissioner of Patents can grant compulsory licences for the use of a patented invention that relates to a food or a medicine, or that is considered "abused" by not being "worked" as defined in the Patents Act.

The following table shows receipts and disposals of applications to the Commissioner for compulsory licences to use patented inventions. Applications filed under Section 67 of the Act were made on grounds of abuse of patent owner. Those filed under Section 41(4) were for licences to import or manufacture patented prescription medicines while those under Section 41(3) to prepare or produce food.

	1978-79	1979-80
Section 67		
Applications received	7	2
Licences granted	1	—
Licences refused	—	—
Applications withdrawn	1	4
Applications pending	8	6
Section 41(4)		
Applications received	33	29
Licences granted	12	22
Licences refused	—	—
Applications withdrawn	3	6
Applications pending	38	39
Section 41(3)		
Applications received	—	2
Licences granted	—	—
Applications pending	—	2

Industrial Designs

The Copyright and Industrial Design Office administers the Copyright Act, the Industrial Design Act and the Timber Marking Act and their Regulations.

The outward appearance of an article of manufacture—its shape, pattern or ornamentation—may be registered as an industrial design. Registration of a new design under the Industrial Design Act gives the registered owner sole rights to use the design in Canada for a period of five years, renewable for one further period of five years.

The following chart summarizes the industrial design transactions of the Copyright and Industrial Design Office during the last two fiscal years:

	1978-79	1979-80
Industrial Designs		
Applications received	2,017	1,818
Designs registered	1,767	1,544
Registrations renewed	1,781	283
Assignments recorded	261	369

Trade Marks

The Trade Marks Office administers the Trade Marks Act and approves and registers applications unless the mark applied for is an apt word for use by all traders or confusingly similar to marks already registered in Canada. A newly registered mark remains on the register for an initial period of 15 years. Before a trade mark is licenced, an application for registered user should be requested from the Trade Marks Office.

In September 1979, the Trade Marks Office embarked on a project to eliminate extensive work backlogs ranging from four to 14 months by April 1, 1980. As a result of this project, the time from the filing date to first examination of a trade mark application has been reduced from eleven and a half months to two weeks; assignment processing from seven and a half months to four weeks; applications awaiting advertisement from five months to one week for date entry; and registered user applications from 14 months to two weeks.

Also, during 1979-80 the Office completed the second phase of an overall plan to convert the majority of clerical functions in the Office from manual to electronic data processing. The information in the *Trade Marks Journal* is now being printed using computer technology. Data already captured during initial processing of new applications is being recalled and updated through computer terminals for publication.

The following table summarizes the Office's transactions over the past three years:

	1977-78	1978-79	1979-80
Trade mark applications filed	14,462	14,781	14,448
Trade mark applications advertised	8,903	8,435	16,327
Trade mark registrations	7,309	5,564	9,905
Registered user applications filed	2,932	3,501	2,696
Number of trade marks governed by registered user applications (registered and pending)	10,481	14,756	11,714
Registered users registered in respect of trade marks	6,975	5,693	18,403
Registered user registrations cancelled in respect of trade marks	2,686	1,610	4,797
Transfer applications filed	5,558	7,882	9,186
Transfers registered	6,672	5,851	9,283
Trade mark registrations renewed	4,061	3,902	3,659
Trade mark registrations expunged	4,611	4,306	3,976
Amendments entered on the register	4,936	3,092	5,379
Copies prepared	330,332	354,995	458,737
Duly registered trade mark agents	5,629	5,970	6,268
Oppositions filed	432	412	488

Technical Advisory Services

In September 1979, it was decided that due to a lack of statutory support the Technical Advisory Services Branch would be abolished on March 31, 1980. During the latter half of the year, Branch operations were phased out and the staff relocated. It had been a major project of the Branch to make patented technical information available to small manufacturers and inventors mainly through provincial research organizations, the Ontario Industrial Innovation Centre at Waterloo University and a number of field offices of the Department of Industry, Trade and Commerce. In 1979-80 the Branch responded to a total of 564 requests for patents relating to specified areas of technical interest.

Research and International Affairs

With the amalgamation of the Bureaux of Corporate Affairs and Intellectual Property in October 1979, the Corporate Research and the Research and International Affairs Branches were combined.

The Branch sought to table legislation amending the Trade Marks Act in Parliament. However this, along with a request for authorization to prepare new legislation amending the Patents Act, could not be considered before Parliament was dissolved. Work is continuing in both areas.

The first draft of a working paper on Industrial Design was completed by the Bureau of Management Consulting, Supply and Services, with the final version scheduled for release in 1980-81.

The Branch prepared briefing material for Canada's participation in the Diplomatic Conference for the Revision of the Paris Convention for the Protection of Industrial Property. In addition, several preliminary meetings of Group "B" countries were convened.

The Branch continued to develop and implement the department's automated name search system, NUANS. Administration of the system is now being assumed by the Corporations Branch. The system's scope and capability has been further enhanced in both corporate names and trade marks.

The Economic Analysis Division of the Branch continued its research on patents with the preparation of a study on the determinants of market power in the pharmaceutical field. This study will include an examination of the role of patents and advertising in the industry.

Research is also under way on the economic effects of patents in the marketplace. As well, the Division examined proposals contained in the draft Industrial Design study and provided economic expertise on related policy issues.

The Division's computerized data base, PATDAT, now contains information on 120,000 patents, including all patents issued in 1979.

News Items

MONACO

Director of Commerce, Industry and Industrial Property

We have been informed that Mr. Etienne Franzi has been appointed Director of Commerce, Industry and Industrial Property.

Calendar

WIPO Meetings

(Not all WIPO meetings are listed. Dates are subject to possible change.)

1982

May 3 to 7 (Geneva) — Permanent Committee on Patent Information (PCPI) — Working Group on General Information

May 10 to 14 (Vienna) — International Patent Classification (IPC) — Advanced Seminar

May 24 to 28 (Geneva) — Nice Union — Committee of Experts

June 7 to 11 (Paris) — Berne Union — Committee of Governmental Experts on Problems Arising from the Use of Computers for Access to or the Creation of Works (convened jointly with Unesco)

June 7 to 18 (Geneva) — Permanent Committee on Patent Information (PCPI) — Working Group on Search Information

June 14 to 18 (Geneva) — Development Cooperation — Working Group on the Establishment of a Guide on the Organization of Industrial Property Activities of Enterprises in Developing Countries

June 28 to July 2 (Geneva) — Committee of Governmental Experts on Model Provisions for the Protection of Expressions of Folklore (convened jointly with Unesco)

September 1 to 3 (Geneva) — Working Group on the Rights of Employed or Salaried Authors (convened jointly with ILO and Unesco)

September 6 to 10 (Geneva) — International Patent Cooperation (PCT) Union — Committee for Administrative and Legal Matters

September 10 (Geneva) — International Patent Cooperation (PCT) Union — Assembly (Extraordinary Session)

September 20 to 23 (Geneva) — Permanent Committee on Patent Information (PCPI) — Working Group on Patent Information for Developing Countries

September 23 to October 1 (Geneva) — Permanent Committee on Patent Information (PCPI) — Working Group on Planning

September 23 to October 1 (Geneva) — Permanent Committee on Patent Information (PCPI) — Working Group on Special Questions

September 27 to 30 (Geneva) — Permanent Committee for Development Cooperation Related to Industrial Property

October 4 to 30 (Geneva) — Revision of the Paris Convention — Diplomatic Conference

October 25 to 27 (Paris) — Berne Union — Working Group on Copyright Questions Connected with the Use of Works by Persons with Defective Hearing or Sight (convened jointly with Unesco)

November 8 to 12 (Geneva) — Working Group on Model Contracts for Licensing or Transferring Copyrights (convened jointly with Unesco)

November 22 to 26 (Geneva) — Governing Bodies (WIPO Coordination Committee; Executive Committees of the Paris and Berne Unions)

November 29 to December 3 (Geneva) — Permanent Committee on Patent Information (PCPI) and PCT Committee for Technical Cooperation (PCT/CTC)

December 6 to 10 (Geneva) — International Patent Classification (IPC) — Committee of Experts

December 6 to 10 (Paris) — Berne Union and Universal Copyright Convention — Working Group on the Formulation of Guiding Principles Covering the Problems Posed by the Practical Implementation of the Licensing Procedures for Translation and Reproduction under the Copyright Conventions (convened jointly with Unesco)

December 13 to 17 (Paris) — Berne Union, Universal Convention and Rome Convention — Subcommittees of the Executive Committee of the Berne Union, of the Intergovernmental Copyright Committee and of the Intergovernmental Committee of the Rome Convention, respectively, on Copyright and Neighboring Rights Problems in the Field of Cable Television (convened jointly with ILO and Unesco)

UPOV Meetings

1982

May 11 to 13 (Salerno) — Technical Working Party for Vegetables

May 18 (Madrid) — Technical Working Party for Agricultural Crops — Subgroup

May 19 to 21 (Madrid) — Technical Working Party for Agricultural Crops

September 28 (Faversham) — Technical Working Party for Fruit Crops — Subgroup

September 29 to October 1 (Faversham) — Technical Working Party for Fruit Crops

October 5 to 7 (Cambridge) — Technical Working Party for Ornamental Plants and Forest Trees

October 12 (Geneva) — Consultative Committee

October 13 to 15 (Geneva) — Council

November 15 and 16 (Geneva) — Administrative and Legal Committee

November 17 (Geneva) — Information Meeting with International Non-Governmental Organizations

November 18 and 19 (Geneva) — Technical Committee

Meetings of Other International Organizations Concerned with Industrial Property

1982

European Patent Organisation: June 7 to 11 and November 29 to December 3 (Munich) — Administrative Council
Hongarian Group of the International Association for the Protection of Industrial Property: September 13 to 17 (Budapest) — Conference
International Association for the Advancement of Teaching and Research in Intellectual Property: September 20 and 21 (Geneva) — Assembly
International Federation of Industrial Property Attorneys: May 10 to 14 (Berlin (West)) — Executive Committee
International League Against Unfair Competition: September 12 to 16 (Bath) — Congress

1983

International Association for the Protection of Indostrial Property: May 23 to 27 (Paris) — XXXII Congress