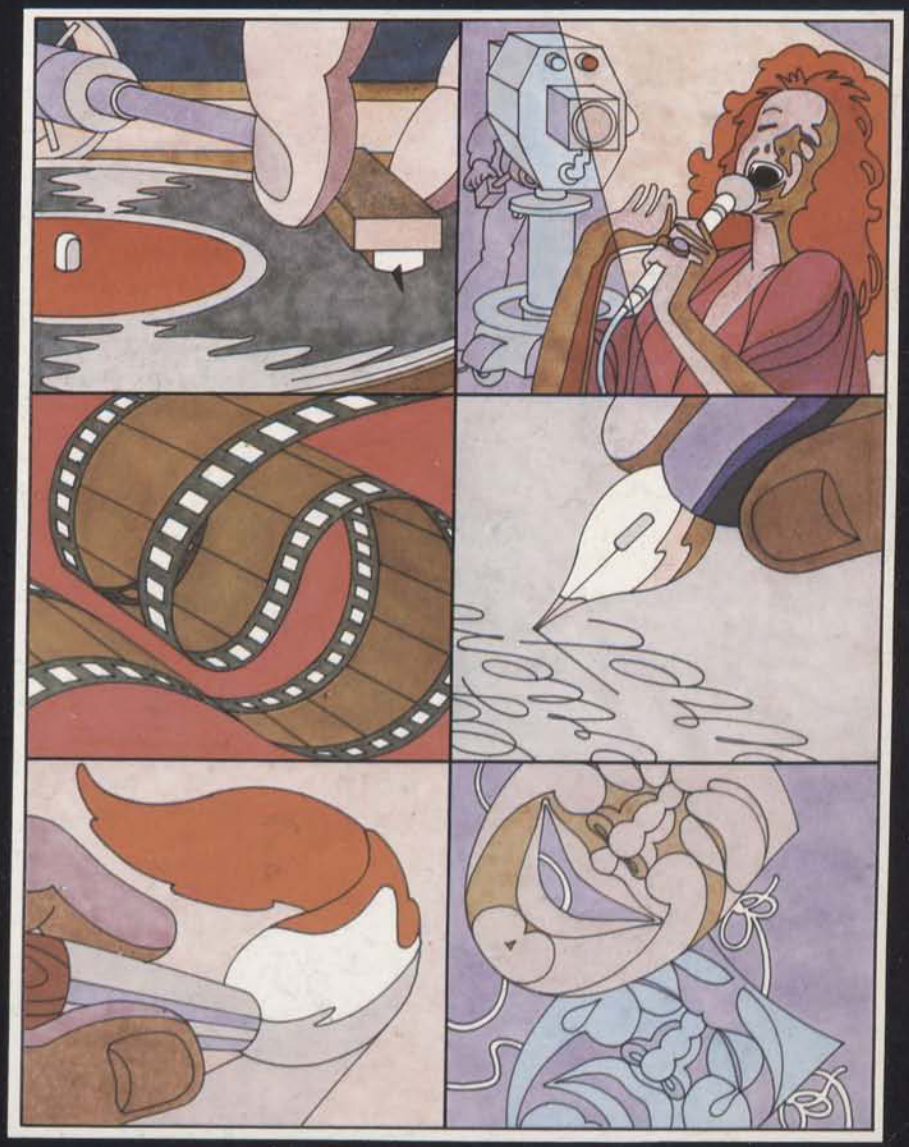
Consumer and Corporate Affairs Canada
Consommation et Corporations Canada

Copyright and the Computer

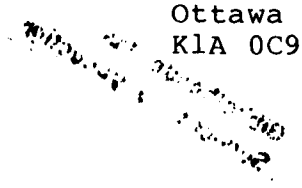
John Palmer
Raymond Resendes



Copyright Revision Studies

En français : Le droit d'auteur et les ordinateurs

Disponible au : Service des communications
Consommation et Corporations Canada
Ottawa (Ontario)
K1A 0C9



COPYRIGHT AND THE COMPUTER

John Palmer
University of Western Ontario

with

Raymond Resendes
University of Western Ontario

Copyright Revision Studies
Policy Research, Analysis and Liaison Directorate
Policy Coordination Bureau
Consumer and Corporate Affairs Canada

The analysis and conclusions of this study do not necessarily reflect the views of the Department.

Copyright Revision Studies
Policy Research, Analysis and Liaison Directorate
Policy Coordination Bureau

Published

Copyright Payment Obligations for Cable Television: Pros and Cons by S.J. Liebowitz, 1980.

The Mechanical Reproduction of Musical Works in Canada by Mike Berthiaume and Jim Keon, 1980.

A Performing Right for Sound Recordings: An Analysis by Jim Keon, 1981.

Term of Copyright Protection in Canada: Present and Proposed by Barry Torno, 1981.

An Economic Analysis of a Performers' Right by Steven Globerman and Mitchell P. Rothman, 1981.

Crown Copyright in Canada: A Legacy of Confusion by Barry Torno, 1981.

Copyright, Competition and Canadian Culture: The Impact of Alternative Copyright Act Import Provisions on the Book Publishing and Sound Recording Industries by Ake G. Blomqvist and Chin Lim, 1981.

The Impact of Reprography on the Copyright System by S.J. Liebowitz, 1981.

Ownership of Copyright in Canada by Barry Torno, 1981.

Fair Dealing: The Need for Conceptual Clarity on the Road to Copyright Revision by Barry Torno, 1981.

Audio and Video Home Taping: Impact on Copyright Payments by Jim Keon, 1982.

Copyright and the Computer by J. Palmer and R. Resendes, 1982.

Available in both official languages from:

Communications Branch
Consumer and Corporate Affairs Canada
Ottawa, Ontario
K1A 0C9

Forthcoming

Exemptions Under the Canadian Copyright Act by D. Magnusson and V. Nabhan.

Remedies and Enforcement by A.M. Butler.

Canadian International Copyright Relations Under the Berne and Universal Copyright Conventions by Barry Torno.

Collective Agencies for the Administration of Copyright by D. Smith.

© Minister of Supply and Services Canada 1982

Cat. No. RG44-1/12E
ISBN 0-662-11748-4

ACKNOWLEDGEMENTS

We owe a deep debt of gratitude to Professor Stan Liebowitz of The University of Western Ontario who made numerous helpful suggestions throughout our work in this study. We also received many very helpful comments from the members of Consumer and Corporate Affairs Canada, particularly Jim Keon and Tony Butler and from Barry Torno of the Department of Justice. Other colleagues at The University of Western Ontario who helped in various ways are Stephen Margolis, Jack Roberts, Jack Quinn, Michael Taggert and John Wren. We would like to thank Marg Gower and her typing pool for their assistance. As well, we appreciate the editorial advice provided by Janet Ritchie. James Fraser was our research assistant.

I initially asked Ray Resendes to provide me with some legal background for this study, but his contribution was so great that it would be unfair simply to list his name in the acknowledgements -- hence his identification as a second author.

I would like to dedicate this study to my children, Matthew, Rachael and Jacob, each of whom is the embodiment of unique expressions of some very fundamental human ideas.

John Palmer
London, Ontario

FOREWORD

This series of studies concerning aspects of copyright law was initiated to provide a better understanding of some important problems and issues involved in the revision of the Canadian Copyright Act. The present Act is now more than fifty years old. The wide breadth of legal, economic and technological developments since the Act was proclaimed underlie the significance of the revision process. The creation and dissemination of information is becoming an increasingly important resource of our society. In addition, the copyright community, including authors, publishers, the film and video industries, broadcasters, the recording industry, educators, librarians and users, contributes hundreds of millions of dollars to the economy. For this reason the Research and International Affairs Branch of the Bureau of Corporate Affairs felt it necessary to undertake in-depth economic and legal research into the cultural, economic and legal implications of the most important of the copyright issues.

With respect to the appropriateness of the economic studies of this series the following passage from the 1971 study of the Economic Council of Canada entitled Report on Intellectual and Industrial Property is perhaps the most perceptive and eloquent:

> It is sometimes implied that where cultural goals are important, economic analysis, with its base associations of the market place, should take a back seat. But this involves a serious misconception of the proper and useful role of economic analysis. It may well be true that in the final analysis, economics is much more concerned with means than with ends, and that the really fundamental "achievement goals" of a society are largely, if not wholly, non-economic in nature. It is also true, however, that, in practice, means can have an enormous influence on ends, whether for good or ill, and that as a result, the systematic analysis of economic means is indispensable both in the specification of social goals and the planning of how to achieve them. In the case of cultural goals, among others, economic analysis can be of great help in bringing about a clearer identification of the goals in the first place, and then in planning for their attainment by the shortest, least costly and most perseverance-inducing route.
>
> It is particularly important that the relevance of cultural goals in a policy-planning situation should not be used as a smoke screen behind which material interests are allowed to shelter unexamined. In an increasingly service-oriented and knowledge-based society, cultural matters in the broadest sense are to a growing extent

> what economic life is all about. They must not fail to be studied in their economic as well as their other aspects. (pp. 139-140)

It is within this spirit that the economic studies completed for the Branch have been commissioned and carried out.

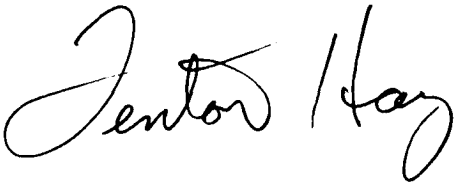
In addition to internal studies, the Branch has contracted with research academics from the Canadian university community who have a special interest in copyright. The external funding of research provides the Branch with new insights and perceptions from some of the most highly skilled academics in Canada with respect to the many complex issues inherent in the revision of the Copyright Act. Additionally, it serves to foster an interest and involvement in these important policy issues amongst others within the academic community. Such involvement and input can only lead to a better understanding and a consequent improvement in the copyright policy formation process.

This study, prepared by Professors John Palmer of the Department of Economics and Raymond Resendes from the Faculty of Law at the University of Western Ontario, examines the computer software industry with respect to the different kinds of intellectual property protection available.

The first part of this study analyzes the costs and benefits of extending intellectual property protection to computer software. It presents, in a clear and lucid fashion, a complete and thorough description of the software industry, the prevailing Canadian case-law and the existing literature on this issue. In addition, Part I defines and clarifies the many critical factors that need to be weighed in reaching a policy decision. Part II explores the impact of the computer on data base production and marketing. It relies in large measure on the analysis developed in Part I. Contained in Part II are discussions of the different types of data bases and an analysis of legal decisions regarding copyright protection for data base compilations. The analysis and overview of these issues again provide a useful backdrop for the development of policy in this area.

Laws and court decisions regarding intellectual property and the computer are continually evolving. No doubt by the time this study is published, new developments will have occurred that deserve serious analysis. One of these is the U.S. Supreme Court decision in early 1981 regarding the patentability of computer software. Although this decision is mentioned in footnote 22 of Part I, Chapter II, the research for and the writing of this study were completed in November 1980.

Given the ever increasing importance of the computer industry in our society this study is especially timely. The thoroughness and rigour with which the issues are examined make this study a valuable research document.



Dr. Fenton Hay
Director
Policy Research,
Analysis and Liaison
Directorate

SUMMARY

Nearly fifteen years ago, Harold Demsetz wrote:

> Changes in knowledge result in changes in production functions, market values, and aspirations. New techniques...invoke harmful and beneficial effects to which society has not been accustomed....[E]mergence of new property rights takes place in response to the desire of the interacting parties for adjustment to new benefit-cost possibilities. (Demsetz, 1967, p. 350)

It is clear that the electronic computer is one of the new techniques which has caused many groups to want to acquire new and additional property rights. This study is a discussion of the computer and two different types of intellectual activity.

In Part I of the study, computer software and its possible forms of protection are analyzed. Having demonstrated that the definition of computer software depends upon what type of protection is to be provided for it and that the type of protection to be provided depends on the definition, it is recommended that computer software be defined as a set of instructions. The software industry has been growing rapidly, particularly among very large and very small firms, and it is pointed out that, with a more extensive form of intellectual property protection, the industry might have grown even faster.

Chapter II of Part I traces the rationales and legal history of intellectual property protection for computer software. It is extremely important that the rationales and legal background be understood; case law is still unclear as to what type of protection is or should be provided for computer software. Additional legislation can probably play a very beneficial role by spelling out more precisely what protection will be available in the future for software, especially if the legislation is worded carefully.

Chapter III examines in detail the costs and benefits of various classes of intellectual property protection for computer software. Patent protection and industrial design protection are rejected in favour of an explicit extension of short-term (25 or 50 years) copyright to cover computer software in addition to the protection currently available in the form of trade secrecy. This recommendation is, however, based on less than satisfactory data. If the estimate is biased against finding net benefits from extending copyright protection to computer software, no net benefits will be found; if the estimate is biased toward finding net benefits, enormous net benefits can be found.

The recommendation is based in part on the technological advances in the computer software industry. Increasingly, the industry is developing general-use packages which are difficult and costly to protect via trade secrecy. Furthermore, software is increasingly being imbedded in such devices as tiny silicon chips or tape cassettes and is then called "firmware." It is argued in this study that:

> It is time for society to recognize that the media for the expression of ideas are continuously expanding as a result of technological change. The law has had to make relatively substantial adjustments to technological progress in the past and will undoubtedly be forced to do so in the future. Composers were once able to receive rewards for their artistic efforts simply by charging a price for printed copies of their music. As the era of sound recordings developed, composers' rewards were seriously jeopardized. Without the extension of intellectual property rights to cover sound recordings, "bootleg" recordings could be mass-produced without payment to the original composers. Yet copyright law had until then addressed only visible copies, not audible copies. A reinterpretation of the scope of copyright was necessitated by technological change to include audible as well as visible copies.
>
> The scope for copyright protection will continue to need reinterpretation as new technologies are discovered and become economically feasible. Even if a particular medium never produces a copy of an expression of an idea which is comprehensible by the unaided human eye or ear, it must still be recognized as a medium for the expression of the idea.

Returning to the quotation from Harold Demsetz cited above, rapid technical change is taking place and is creating a great deal of uncertainty in society as to who does and should have many property rights. The recommendation that copyright protection be extended to cover computer software arises from a numerical analysis of comparative costs and benefits and from the expectation that new technologies will create increasing benefits for society if these additional property rights are created.

In Part II, the impact of the computer and of information storage and retrieval systems (ISRS) on data base production and dissemination is analyzed. It explores the impact of the computer on data base production and marketing. The role of the computer raises many interesting issues with which economic units producing and marketing data bases have had and will have to grapple. These issues have been and

will continue to be dealt with quite effectively by those units or by others entering the industry. There seems generally to be no need to change the copyright legislation concerning data bases.

Part II begins with a discussion of different types of data bases and it is pointed out that intellectual property protection is important for only certain types of data bases. A consideration of the legal history of copyright and compilations shows that a reasonably clear and efficient property right for data bases has already been created, regardless of whether they are used with the computer. There is no need to change the status quo in this case.

There is more than just the advent of the computer which unifies the two parts of the study. They are intrinsically linked by the underlying rationales for intellectual property protection. In both parts of the study, it is emphasized time and again that an underlying rationale for intellectual property protection is that it enables people who create new ideas, products, processes or expressions of ideas to capture more of the benefits of their creative activity. From an ethical point of view alone, protection seems fair and justifiable. From an economic efficiency point of view, as well, intellectual property protection is often justified. The question addressed here is: Given the technological changes created and enabled by the computer what, if any, changes should be made in the intellectual property protection for computer software and for computer data bases?

There is a standard methodology in economics for answering such questions. The first step in this study was to explore whether, in each case, there is a market failure relative to the constant-cost, perfectly competitive norm. For both computer software (particularly recent types of software) and data bases such a market failure was found to exist. The presence of such a market failure is not, however, by itself a justification for government interference in the marketplace. Government policy is not costless. The creation of new and additional intellectual property rights would, for example, impose costs on society. Only after these costs are examined can it be determined whether and how such rights should be created.

TABLE OF CONTENTS

LIST OF TABLES AND FIGURES

Figures

PART I

COPYRIGHT AND COMPUTER SOFTWARE

INTRODUCTION

Whether computer software should receive legislative intellectual property protection has been a question of public policy concern for more than a decade. In 1971, the Economic Council of Canada recommended that no legislative action be taken regarding computer software. A similar recommendation was made by Keyes and Brunet in 1977. In the same year, in the United States, the Computer Software Subcommittee of the Commission on New Technological Uses of Copyrighted Works (CONTU) recommended in a majority report that computer software be granted copyright protection. Similar recommendations were made by Niblett and Anderson in their report to the Whitford Committee in the United Kingdom. And, arguing that copyright might have some limitations, the World Intellectual Property Organization (WIPO) proposed in 1978 that its member countries adopt new legislation to protect the intellectual property of computer software.

Rough estimates of the social costs and benefits of providing intellectual property protection for computer software indicate that, for traditional computer software and ignoring international considerations, the net social benefits of copyright or similar protection would be at least $20,000 per year. These benefits are small, and the margin for error in the calculations is large enough that, despite efforts to stack the deck against finding any social benefits, little confidence can be placed in the positive sign of the estimates. Thus, it is impossible to reject with confidence the hypothesis that it would not make much difference to Canada whether traditional software were protected by copyright, by new and similar legislation, or not at all. If it made no difference, then no new property rights should be created.

The overall conclusion is, however, somewhat different because of the development of new software in the form of magnetic tape cartridges and read-only-memory silicon chips, technology which has come to be called firmware. Status quo methods of recuperating the development costs of traditional software will not be so successful with firmware due to its potential for widespread use. The inclusion of firmware in the consideration of social costs and benefits of alternative modes of protection for software may increase the net social benefits of copyright or similar legislation sufficiently that Canada would benefit from extending such protection to software both in its traditional form and as firmware. This conclusion cannot be put forward with complete confidence, however, because of uncertainty about the extent to which Canada's relatively small size in the world market for computer software reduces the benefits as estimated here.

Chapter I

LEGAL PROTECTION OF COMPUTER PROGRAMS FROM UNAUTHORIZED USE

The Definitional Problem

The importance of the definition. In order to gain a clear picture of the issues that have emerged in attempts to apply legal remedies to the protection of computer programs, a description of the subject matter of computer programs and the process by which they are generated must first be set out. As will be illustrated later, much of the confusion and uneven development of the law in this area is attributable to the lack of uniformity in defining and characterizing this new type of "property." This lack of uniformity is not surprising. The elusiveness of a precise and uniform definition of computer programs arises from the peculiar nature of the product, which can reasonably be characterized on several levels.[1] This problem of definition is crucial. The applicability of statutory means of protection such as copyright or patent to computer programs will depend to a great extent on how the courts view the essential nature of the programming process. Controversy abounds on the issue of the proper characterization of computer programs. Unless some consensus emerges, the kind and extent of legal protection available for computer programs will continue to be clouded by their present uncertain scope and ad hoc development. This definitional problem is further complicated by the increasing interchangeability between computer hardware and software; many of the functions performed by software programs can now be built into computer hardware.[2] This "hardwiring" of programming functions has led to a new concept of a hybrid product known as firmware.[3]

---

1. An excellent discussion of the definitional problems associated with computers and technology is presented in Popper, 1977.

2. For a detailed explanation of the interchangeability between programs in software form and programs in "hardwired" or "firmware" form see Ross, 1978.

3. One commentator has recently noted that:

> the rapid advancement of computer technology is adding a new uncertainty....This uncertainty lies in the field of "firmware," specifically the issue whether courts will hold that firmware is patentable....A traditional dichotomy has been drawn in computer science between "hardware" and "software." Hardware can be defined as the collection of physical components and apparatus that make up a computer system, whereas software consists of information (data) and instructions for processing this

The following section briefly describes the essential steps in the programming process and examines alternative definitions and characterizations that have been applied to the end product.

The programming process. A computer program has been most commonly defined, in general terms, as a "set of precise instructions that tells the computer how to solve a [particular] problem."[4] It is useful to distinguish at the outset between two broad classifications of programs: operating systems programs and source programs. Almost all computers have built-in operating systems programs. Reduced to its simplest terms, the essential hardware component of a computer consists principally of an enormous assembly of switches which, at any particular time, are either off or on. In order for the computer to perform a particular task on data fed into it or stored within it, a mechanism for actuating these switches according to specified instructions is necessary. The initial setting of switches, either by the computer manufacturer or with a different program, constitutes an operating systems program. This program controls the operation of the physical machinery (hardware) by converting particular instructions, fed by means of a source program, into machine language which in turn sets in motion a series of electrical impulses enabling the computer to perform the required task.[5] The hardware, together with the built-in operating sys-

---

> data (programs) fed into the computer, on which it operates. This dichotomy however, has not withstood the assault of technology. Occupying an amorphous middle ground between hardware and software is "firmware," technically defined as "microprograms resident in the computer's control memory"....A generalized example of firmware would be a small, easily replaceable part which is pre-punched or wired in such a way that when activated...uses a particular sequence of computer operations to take place. (Ross, 1977, pp. 735-737)

Programs carried out by means of integrated circuitry on a silicon chip are another example of firmware. The degree of fixation and ease of modification or alteration of programs embodied in firmware form can vary significantly. Use of firmware is becoming increasingly widespread in the industry because of its cheapness and flexibility.

4. This is the most common definition but it is by no means universally accepted as the correct characterization of computer programs. Alternative definitions will be presented later in the study. See also footnote 6.

5. This is a highly simplified explanation of how a computer operates but it is sufficient for an understanding of the issues that this study will address.

tems program, defines the capability of the computer and the flexibility afforded to the programmer in devising source programs to solve specific problems. Operating systems programs, unlike source programs, are an integral part of the mechanical structure of the computer and cannot be read without the aid of special equipment.[6]

A source program is the ordinary type of program written by a programmer to deal with a particular task to be solved. In general, the development of a source program consists of several steps. First, the problem to be solved must be clearly formulated. This is done by means of a flow chart, which is essentially a schematic representation of the underlying logic of the program -- commonly referred to as the program's algorithm. An algorithm is a precise characterization of a method of solving a problem, including a detailed breakdown of the task into its essential elements and a specification of the sequence of steps to be followed by the computer to reach the solution (Ralston and Meek, 1976, pp. 47-48). Since a computer cannot read a flow chart in diagrammatical form, the next step is to reduce the algorithm and to express it in a language that the computer can understand. The flow chart is generally translated into a standardized computer programming

---

6. In Data Cash Systems Inc. v. JS&A Group Inc., Judge Flaum distinguished between a source program and an object program in his definition of a computer program:

> A computer program has been defined generally as a set of precise instructions that tells the computer how to solve a problem. C.J. Sippl and C.P. Sippl, Computer Dictionary 333 (2d ed. 1974); Synercom Technology, Inc. v. University Computing Co., 462 F. Supp. 1003, 1005 (N.D. Tex. 1978). Normally, a computer program consists of several phases which may be summarized as follows. The first phase is the development of a flow chart which is a schematic representation of the program's logic. It sets forth the logical steps involved in solving a given problem. The second phase is the development of a "source program" which is a translation of the flow chart into computer programming language, such as FORTRAN or COBOL. Source programs may be punched on decks of cards or imprinted on discs, tapes or drums. The third phase is the development of an "assembly program" which is a translation of the programming language into machine language, i.e., mechanically readable computer language.
>
> Unlike source programs, which are readable by trained programmers, assembly programs are virtually unintelligible except by the computer itself. Finally, the

language such as BASIC or FORTRAN.[7] At this stage, the source program resembles coded writing or nonsensical literary prose. The computer cannot read the program, however, even in this form and the source program must be punched on cards or imprinted on disks, tapes or drums which, when fed into the computer by means of an input device such as a card reader or terminal, are translated into machine language by the operating systems program.[8] The source program acts as a catalyst with respect to the built-in operating system program and triggers the setting of switches in a manner designed to enable the computer, by utilizing the logical sequence of steps set out in the source program, to solve the particular task.

This brief description of the development and function of a computer program is intended to explain the conceptual problem of defining its subject matter for the purpose of legal protection. The problem arises because, at each stage of the program development and use, the

---

> fourth phase is the development of an "object program" which is a conversion of the machine language into a device commanding a series of electrical impulses. Object programs, which enter into the mechanical process itself, cannot be read without the aid of special equipment and cannot be understood by even the most highly trained programmers. J. Brown and R. Workman, How a Computer System Works 149-175 (1976); Keplinger, Computer Intellectual Property Claims: Computer Software and Data Base Protection, 1977 Wash. L. Q. 461, 464; M. Pope and P. Pope, Protection of Proprietary Interests in Computer Software, 30 Ala. L. Rev. 527, 530-31 (1979).
>
> Thus, at some point in its development, a computer program is embodied in material form and becomes a mechanical device which is engaged in the computer to be an essential part of the mechanical process. At different times, then, a given program is both "source" and "object." The "source program" is a writing while the "object program" is a mechanical tool or machine part. (Data Cash at E-1)

It will be demonstrated later why Judge Flaum's distinction between source and object programs should not be accepted.

7. There are many programming languages. BASIC and FORTRAN are the most commonly employed as they have achieved a great deal of standardization.

8. More specifically, the translation of the source program is effected by a device known as the compiler, which is part of the operating systems program. A printed program in machine language would appear as nothing more than a series of zeroes and ones.

form and, in some cases, the substance of the program undergoes radical alterations.

What constitutes the program? The constructed flow chart used to express the underlying algorithm? The "literary" content expressed in the form of translation of the flow chart to a computer language such as BASIC or FORTRAN? The configurations of holes punched on cards or imprints on tape or disk which, when fed into the computer, become an integral part of the mechanical process within the computer? The following section presents a number of the alternative definitions and characterizations of computer programs that have been formulated by commentators, the courts, and government committees. As will become clear later, although each of the alternative definitional approaches is reasonable, each carries radically different legal consequences.[9]

Possible definitions

(a) A computer program is synonymous with an algorithm. The proponents of this view equate computer programs with their underlying algorithms. For example, one noted expert in the field adopts this definition:

> "Algorithm" is synonymous with "computer software" or "computer program." With the advent of high speed digital computers in the early 1940's the word "algorithm" took a slightly different meaning....In many discussions the word simply means a computer program. (Anderson, 1975, p. 369)

A similar proposed definition is that "a [computer] program is the expression of an algorithm in programming language" (Horowitz and Sahni, 1978, p. 2).[10] This approach views the computer program as merely a particular format for the alternative description of the algorithm which has first been graphically depicted by means of a flow chart. If one accepts this approach to characterizing computer programs as the most appropriate, the potential number and kinds of legal remedies available against unauthorized use become extremely limited. If the only real novelty and utility of programs lies in their underlying algorithms rather than in what forms the programs take, then they will not fit within the traditional scope of copyrightable subject matter. Furthermore, if what is sought to be protected in a program is merely the

---

9. This is because the applicability of statutory forms of protection such as patent and copyright depend on the classification of the subject matter in question.

10. Note, however, that a program is defined as an expression of an algorithm. This distinction between an expression of an algorithm and the algorithm itself is of paramount importance, as will be seen in the section on copyright protection.

logical sequence of steps, which is arguably the essence of a computer program, then patent protection may also be inappropriate. If algorithms are merely ideas,[11] scientific principles, or abstract theorems or methods which depend for their novelty solely on a series of mental or intellectual steps, then, if one equates programs to algorithms, they will be per se unpatentable. Hence, the applicability of statutory forms of protection, such as copyright and patents, would require radical and substantive legislative change in order to encompass computer programs within their protective scope. Barring such substantive changes in intellectual property statutes, the programming industry would have to rely solely on the common law doctrine of trade secrets and technological means of protection.

(b) Computer programs are sets of instructions expressed in a precise, tangible form. The notion of a computer program as a detailed set of instructions which is fixed in a tangible medium and intended for the operation of a computer is perhaps the most widely accepted definition. The U.S. Supreme Court in Gottschalk v. Benson (409 U.S. 63 [1972] at 65), for example, defined a program as "a sequence of coded instructions for a digital computer." A similar definition advanced by the U.S. Copyright Office reads as follows:

> In general, a computer program is either a set of operating instructions for a computer or a compilation of reference information to be drawn upon by the computer in solving problems. In most cases the preparation of both of these types of programs involves substantial elements of gathering, choosing, rejecting, editing and arranging material. Some types of programs also embody verbal material which is written by the programmer and could be considered literary expression. (Cary, 1964, p. 362)

Still another proposed definition is contained in section 3(a) of the Federal Computer Systems Protection Act of 1979:

> Computer program means an instruction or statement or a series of instructions or statements in a form acceptable to a computer which permits the functioning of a computer system in a manner designed to provide appropriate products from such computer system.

In Canada, the Patent Appeal Board has adopted a similar definition:

---

11. Some commentators have argued that an algorithm is neither a law of nature nor a disembodied idea but rather a physical manifestation of an idea and is therefore proper subject matter for patentability when tied to a specific device or application. See M. Novick and H. Wallenstein, 1980. The courts have tended, however, to equate an algorithm with an abstract mathematical formula or relationship. See, for example, Gottschalk v. Benson 409 U.S. 63 (1972), p. 65.

> The term program is taken to mean a set of ordered steps or list of instructions....This set of steps or list of instructions may be recorded on a variety of media including printed or handwritten lists on paper, punched cards or paper tapes, magnetic tapes or electric wiring. (Patent Office Record, August 1, 1978, p. 27)

Finally, the software subcommittee to the National Commission on New Technological Uses of Copyrighted Works (CONTU) adopted the following definition for the purpose of copyright law: "A fixation of a series of statements or instructions to be used in conjunction with a computer to bring about a certain result" (1977, p. 16).

Many of the proponents of this viewpoint are, not surprisingly, in favour of copyright as the appropriate statutory mechanism to protect interests in computer programs.[12] This definitional approach stresses that the program's form, as expressed in language or symbols on a particular medium, has independent utility apart from the program's underlying algorithm in the sense that improper translation from flow charts to computer language could result in total failure by the computer in arriving at a solution. Hence, the proponents of this view argue, copyright is entirely appropriate if restricted to protect an expression of a logical sequence of steps in an original manner as opposed to the underlying logic per se (CONTU, 1977, p. 26).

(c) A computer program is a process which directly controls and regulates the mechanical functioning of the computer in a desired manner. A number of definitions characterize programs in terms of their ability to control the actual computer hardware. For example, "a computer program is basically a system that controls the activity of the computer" (Banzhaff, 1964, p. 1276).

---

12. The final CONTU report submitted to Congress in July 1978 recommended that copyright protection be explicitly extended to computer programs. The World Intellectual Property Organization (WIPO) concluded in Model Provisions on the Protection of Computer Software that programs require a special type of legal protection since they do not fit easily within the traditional ambit of existing statutory forms of protection such as patents or copyright. The Model Provisions set out a suggested statutory scheme for the protection of computer programs which is similar in substance to copyright but provides for a shorter term of protection and prohibits both unauthorized use and disclosure as well as the copying of programs. WIPO defines a computer program for the purpose of its model act 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." (Section 1(i). The Model Provisions are reproduced in Appendix B of this report.)

Similarly, "generally speaking software consists of the programs which control the computing machinery" (Davis, 1979, p. 1).

The proponents of this definitional approach regard computer programs not as just a set of operating instructions for solving particular problems on the computer but rather as an integral part of the computer's machinery. The computer is incapable of functioning in its intended manner until and unless it has been programmed. Programs are intended to set the computer's internal switches and enable it to perform the required task. Hence "...the programming sets the switches, in effect redesigning the internal structure of the machine and becoming an inseparable part of the machine. The program may thus be viewed as a machine part or as the completion of a previously incomplete machine" (Gemignani, 1980, p. 279).

Many of the proponents of patent protection for computer programs adhere to this viewpoint. They argue that patents would protect not the novel underlying algorithm but rather the novel operation of the physical computer hardware that is constrained to operate in a particular, unobvious manner by the program.[13]

(d) The importance of the definitional problem. A computer program is a unique type of intellectual property which can reasonably be defined in several manners. Indeed, computer programs may be viewed not so much as a product per se but rather as a process for achieving a particular result on a computer. Although many of the definitional distinctions may be thought of as merely semantic, the characterization of computer programs in one manner as opposed to another has determined and probably will continue to determine the kind and extent of legal remedies available to and/or sought by developers of software in protecting their proprietary interests. Hence, in a very important sense, the issue of legal protection for computer programs revolves essentially around the problem of definition.

Chapter II examines in detail the particular problems that have arisen in applying legal remedies to the protection of computer software. As a survey of the case history will demonstrate, the courts in the United States and Canada have exhibited an extreme uneasiness in attempting to fit computer programs into the traditional ambit of statutory forms of protection such as patents and copyright. This is partly due to the troublesome definitional problems outlined above and partly due to the courts' lack, for the most part, of the institutional competence to deal with the complex and constantly developing technology of computers and computer software.

---

13. This is the view adopted by the U.S. Court of Customs and Patent Appeals. See the discussion of patents and computer programs later in the text.

Recommended definitions

(a) Software as part of a process or machine. Those favouring this definition point out that a computer program sets and alters a large number of switches in a computer. They argue that computers with switches set in different positions are, in reality, different machines, with different functions and generating different outputs. This argument can be rejected on the grounds that it is inconsistent with traditionally accepted definitions of a machine and that it misrepresents the roles and functions of software and hardware.

The inconsistency of the argument can be seen most clearly by examining many multifunction machines. For example, a camera which can take pictures at different exposure speeds is simply that -- one camera with some built-in flexibility. Different cameras could be built with different fixed exposure times but the flexible camera is just one camera, regardless of which exposure time is used by the photographer. Similarly, in a sound reproduction system the electrical current flows through the system differently for each different sound recording which is being played. Nevertheless, the sound reproduction system is not considered as a different system for each sound recording. Instead, like the camera, it is considered as a single system with, in this case, the flexibility of playing different recordings. A computer is similar to these and other flexible-use machines or systems. The software does not become part of the mechanical system; rather, the system is flexible enough to accept a broad range of different programs.

This understanding of computer systems leads to the second grounds for rejecting the definition of computer software as part of a machine. Software is not inextricably entwined with a specific machine. It is a set of instructions which can be used on different machines. Each machine is capable of using different software. Because the software and hardware are not uniquely linked, the software cannot be considered to be part of the machine.

Although the definition of computer software as part of a machine can readily be rejected, it is less easy to reject it as part of a process. Computer software is essentially a set of instructions, a series of steps to be taken by a computer. Certainly these instructions or steps can also be viewed as part of a process. As will be shown in the discussion of the patentability of computer programs in Chapter II, however, the U.S. courts have (correctly) traditionally rejected process patent applications for computer software, invoking a principle known as the mental steps doctrine. Because this paper agrees with the courts' position, computer software is defined here in terms which do not encourage construing it as qualifying for process patents, even though the set of instructions embodied in the software may well form a part of a process which does qualify for patent protection.

(b) Software as a set of instructions. Computer programs are usually extended sets of instructions to a machine to perform addition, subtraction and conditional functions. There is no reason to imbue software with more characteristics than it really has and most experts agree with this definition. Although many similar definitions have been presented above, the WIPO definition is perhaps most suitable: "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" (WIPO, 1978, p. 9). Adopting the WIPO definition would have the added advantage for Canada of standardizing its definition with that of other countries also adopting this definition. Alternatively, since the bulk of Canadian trade in computer software is most likely to be with the United States, Canada may find it more advantageous to adopt a definition like that recommended by CONTU: "a fixation of a series of statements or instructions to be used in conjunction with a computer to bring about a certain result" (CONTU, 1977, p. 16).

There is very little difference between the two definitions. The CONTU definition is less wordy and is perhaps preferable for that reason alone. As well, because it defines software as a fixation, it can perhaps more easily be interpreted as including software which is mass-produced as firmware.

(c) Software and firmware. With integrated circuits and miniature silicon chips, it is possible to embed computer programs in very small microprocessing units. Software can also be produced on magnetic tape or tape cartridges. These are examples of what has come to be known as firmware.

There should be an increasing opportunity for developing firmware markets over the next decade. Although this type of software is usually designed and developed using traditional methods involving programming and hardware, it becomes fixed in chip or tape form for marketing purposes. Firmware, in this sense, is analogous to sound recordings, which embody a traditionally protected intellectual property in a form which was not protected at the turn of the century.[14] It is recommended that the legal definition of fixation, whether for copyright or for new legislation covering software, be expanded to include fixation of software in firmware. Firmware is a new medium for the expression of ideas, much the same as sound recordings were new media for performances of copyrighted musical compositions.

---

14. Music recorded on sound recordings was not protected by copyright in the United Kingdom until 1911. However, the 1911 Act did not apply in Canada and music reproduced in sound recordings was not protected by copyright here until 1921. See Fox, 1967, pp. 30-34, 178-179 and 186.

Summary. In summary, the way in which computer software is defined will affect the type of policy recommendations proposed for the legal protection of software. Conversely, the results of a cost-benefit examination of various policies will affect what recommendations are made concerning the appropriate definition of software. This paper rejects the definition of computer software as part of a machine by analogy to other flexible-use machines. The decision to recommend the definition of software as a set of instructions follows in part from economic analysis which indicates that there is a strong likelihood that there would be positive net social benefits from creating copyright protection of software. Before presenting this analysis, however, this chapter concludes with a discussion of the structure of the Canadian computer software industry. Chapter II describes the present legal protection environment as a background for this paper's analysis and recommendations.

The Structure of the Canadian Software Industry

There are two common images of the software industry. On the one hand is the image of many small firms composed of one or two people who enjoy the intellectual challenge of computer programming, operating out of small environments and earning considerable psychic rewards in addition to their pecuniary incomes. On the other hand is the image of giant, integrated firms which manufacture computer hardware with ancillary business involvements in software. The truth embodies each of these images along with others between these two endpoints on a spectrum.

Computer software. As pointed out in the previous section, the definition of computer software is moot. The various definitions of software all embody, however, a central notion that software encompasses the different types of computer programs. In this broad context, programs are designed to set electrical switches in computer hardware and can be embodied in many different forms. They can be punched in cards, recorded on tapes or discs or fed into microprocessing units on silicon chips. Regardless of the media via which they are transmitted and implemented, all forms of computer programs are included within the scope of the definition of computer software for the purposes of this study of the industry's structure.

Data dealing specifically with the software industry in Canada are difficult to come by. Statistics for large firms are often difficult to disentangle from their activities in numerous diverse but related non-software business activities. Furthermore, official government statistics for large firms are sometimes unreported due to legal proscriptions against the disclosure of confidential information provided to Statistics Canada. Statistics for small firms are just as difficult to obtain. Many small firms are not even included in a survey because they do not come to the attention of the data collectors. And frequently, in small firms, each employee performs several different functions, thus making it dif-

ficult to determine how much of each firm's resources is devoted to any specific type of activity.

This study has used two different data sources. The first is the annual computer services survey published by Statistics Canada (cat. no. 63-222). The second is an annual survey published in Canadian Datasystems from 1972 to 1978, compiled initially by Urwick, Currie, and Partners Ltd. and amended by Canadian Datasystems.[15] Each survey has considerable strengths and weaknesses; fortunately many of the strengths of each survey complement those in the other and so in-depth analysis of each will help to provide a more complete picture of the industry.

Statistics Canada survey. The Statistics Canada survey compiles data not only for the software industry but for the entire computer services industry. The data are collected for each establishment rather than for each firm.[16] The survey defines computer services to include processing services, input preparation, software and systems services, systems development and maintenance, other software and systems services, equipment maintenance services and other computer services (including computer-related education services, computer facility management and feasibility studies). As is clear from this definition, the Statistics Canada survey includes establishments and activities not involved with the development and marketing of computer software. For many years and for several of the variables studied, however, the data do permit a finer distinction than is implied by the definition. The survey may be overly broad for the purposes of this paper in another respect: it includes establishments "reporting that...receipts from sale of computer services are greater than one-third of total operating revenue." In other words, establishments for which nearly two-thirds of operating revenue are derived from other sources are included in this survey of the computer services industry. In fact, in the first two years of the survey this distinction was not so carefully drawn and many establishments deriving more than two-thirds of their operating revenue from, say, the sale, lease and/or rental of hardware and equipment were included. These cautionary considerations should be kept in mind in the following examination of the data provided in the Statistics Canada surveys.[17]

---

15. A broader but considerably less detailed survey in the January 1979 and 1980 issues was also used.

16. An establishment is a geographical and physical entity, whereas a firm is a business or economic entity. A firm may be composed of more than one establishment if it is what is commonly referred to as a multiplant firm.

17. Except where noted, the basic data upon which the ensuing discussion is based are presented in Appendix A.

(a) Number of establishments. There has been very rapid growth in the number of establishments in the computer services industry. In 1972 there were 309 establishments and in 1978 there were 698. The increase in the number of establishments during this period may have been due in part to expanded coverage by the Statistics Canada survey. However, most of this increased coverage, to the extent that it occurred, would probably have been among smaller establishments. Indeed, as is shown in Table 1, the number of establishments in the smallest size class has grown dramatically, from 147 to 282. As a percentage of all establishments, though, those with less than $100,000 in operating revenue have dropped from 48 per cent to 40 per cent. The number of establishments in the smallest size class less than doubled, while the number of establishments in all of the larger size classes more than doubled. Surely one explanation for these results is that with the rapid inflation during the 1970s, each size class contained smaller establishments over time. In general, then, all sizes of establishments grew in number over time, but the larger establishments grew in number more rapidly than did the smaller ones.[18]

Despite whatever concerns there may be about the reliability of the data reported in the Statistics Canada survey, they present a picture which is not very different from the size distribution of firms (not establishments) in the United States which are members of the Association of Data Processing Service Organizations (ADAPSO). Of course, membership of ADAPSO might not necessarily constitute a random sample of all computer service firms in the United States but there is no strong reason to believe that the numbers reported in Table 2 are seriously biased in any particular direction. The size distribution of firms appears to be quite similar to that of the Canadian establishments, although no tests for significance are possible since the boundaries on the size classes are different.

---

18. Two forces may have been working to create the reported growth rates of smaller establishments. The first is the increased coverage of the Statistics Canada survey, as mentioned in the text. The second is that some of the more successful larger firms set up branch establishments in various cities. These branch establishments often fell into the smallest size class even though they were part of a larger firm. Evidence that this phenomenon existed is not available from Statistics Canada surveys but is indicated in Frank, 1979. See also the Canadian Datasystems surveys.

Table 1

Number of Establishments by Size Class

	Size classes by $1000s total operating revenue				
	less than 100	100-499	500-1999	more than 1999	total
1972	147	95	46	21	309
1973	142	112	47	30	331
1974	-	-	-	-	345
1975	144	160	63	30	397
1976	198	179	79	33	489
1977	246	215	88	47	596
1978	282	256	108	52	698

SOURCE: Appendix A

Table 2

ADAPSO* Membership by Size of Firm in 1977

Revenues	Number of members
$100,000 - 1 million	201
$1 million - 5 million	75
$5 million - 10 million	13
$10 million - 50 million	12
over $50 million	7

SOURCE: ADAPSO, 1978(b)

*Association of Data Processing Service Organizations

(b) Chaos, competition, and lognormality. The prevailing view of the computer services industry is that it is dominated by chaotic competition: firms enter and exit rapidly and in large numbers and each expects to dominate the field in its specialization within two or three years.[19] This type of chaos is characteristic of competitive industries and usually leads to a lognormal size distribution of firms (i.e., one which looks like the bell-shaped normal distribution when logarithms are taken of the sizes). The statistical theory underlying this phenomenon is known as "Gibrat's Law," characterized as follows:

> stochastic growth processes adhering to Gibrat's Law generate a log normal size distribution of firms -- that is, a distribution highly skewed when sales are plotted by the frequency of their actual values, with one or a few firms realizing high sales while most make low sales, but which is normal and symmetric when logarithms of firms' sales are plotted. (Scherer, 1980, p. 147)[20]

There are too few size classes to permit a rigorous test of whether the observed size distributions of firms presented in Tables 1 and 2 significantly differ from a lognormal distribution. Nevertheless, it appears that the size distributions do coincide with the general characteristics of a lognormal distribution -- namely, many small firms or establishments and few large firms or establishments. Consequently, it seems highly plausible that the present size distribution of firms is the result of a random growth process via which a few lucky firms became large while most firms remained small. This means that although there is not an even distribution of market shares among all of the firms in the industry, the structure of the industry is nevertheless likely to be competitive.

(c) Employment. It was suggested earlier in this section that, due to the relatively high rate of inflation during the 1970s, one might expect that the number of firms in the smallest size class would not have increased as rapidly as the number of firms in larger size classes. This suggestion is based on the idea that a firm with the same inputs and outputs in real terms could have moved from one size class to the next as its inputs and outputs became valued more highly in nominal terms. An examination of employment trends over time by size class yields data consistent with this idea. Generally, it appears that the number of employees (incuding working proprietors) has declined in each size

---

19. See, for example, Frank, 1979; Soma, 1976; Share Inc., 1976; Brock, 1975.

20. A very interesting, thorough, yet concise discussion of random growth processes and size distributions is presented by Scherer, 1980, pp. 145-150.

class.[21] This decline was likely due, in part, to firms remaining the same size in real terms but being classified as larger firms as revenues increased with the rate of inflation (see Table 3). If, however, the inflationary effects had accounted for all of the decline in the number of employees by size class and if all firms had remained the same size in real terms, the average number of employees for all firms should not have declined, as it actually did. (Note especially the decline in the average number of employees for all firms between 1975 and 1978, years for which the data are more directly comparable.)

Table 3

Average Number of Employees
(Including Working Proprietors) per Establishment

	Size classes by $1000s total operating revenue				
	less than 100	100-499	500-1999	more than 1999	all establishments
1972	3.62	14.24	52.83	480.81	46.64
1973	4.05	14.13	43.96	331.67	42.82
1975	3.09	11.68	44.60	153.27	24.42
1976	2.88	10.53	38.57	144.82	21.04
1977	2.34	9.71	35.35	129.91	19.93
1978	2.23	8.16	31.24	136.63	18.91

The decline in the average number of employees for all firms has resulted from several different market forces. First, there has been a technological change in the computer services industry such that firms can now produce output of the same real value with fewer employees. It does not seem at all unreasonable to expect that labour productivity would have increased in the computer services industry

21. There was a fairly dramatic decline in employment in the largest size class between 1973 and 1975 because, in 1974, Statistics Canada began excluding from this portion of the survey firms which derived less than one-third of their revenues from computer services.

during the 1970s. And, as technological growth was occurring, many of the less efficient firms would have been driven out of business, leaving more efficient firms with higher productivity in the industry. Exit by less efficient firms could give the impression of technological change in the industry when, in reality, market forces created that impression.[22] In addition, there has been a growth in the market for package as opposed to custom software. A survey of more than 300 clients of Informatics revealed that, from 1974 to 1977, "the percent of new applications implemented with outside purchased software for each of the four years of the survey was 4%, 7%, 10%, and 12%, respectively. For non-U.S. users, who were surveyed beginning a year later, the percentages were 3%, 7%, and 12%, respectively" (Frank; Jan. 15, 1979; In Depth p. 11).[23] Using a different data source, Business Week estimates that sales of software packages will be greater than the sales of custom-developed software in 1980 and should nearly double the sales volume of custom software by 1985. This phenomenal growth in package software sales will have substantial importance for a decision about the appropriate legal protection for software. Because there has been a growing market for packaged software, firms in the computer services industry will likely need fewer programmer employees to market the same dollar volume of previously developed programs.

(d) Software vs. computer services. As emphasized earlier, the Statistics Canada data are from a survey of computer services establishments. Not all computer service firms offer programming, however, and nearly all computer service firms sell more than just programming services. In 1974, for example, 66.7 per cent of service operating revenues were generated by processing services (over-the-counter, remote access, communication line charges and input preparation services), while only 24.0 per cent of the revenues were generated by software and systems services. The percentage of operating revenue derived from software and systems services is presented in Table 4 for 1972 to 1978 for all size classes of establishments.[24] From 1975 through 1978 there appears to have been a slight increase in this percentage overall. No clear pattern emerges as to which size class emphasized software more over time, with the possible exception of the smallest size class where the percentage increased from 40 per cent to 49 per cent from 1975 to 1977.

---

22. The discussion of Canadian Datasystems surveys, below, will shed some light on this question.

23. A similar trend appears in Business Week, September 1, 1980, p. 54.

24. In 1974 Statistics Canada changed its definitions and its criteria for inclusion in the survey. Hence, the numbers change somewhat dramatically between 1973 and 1975.

Table 4

Software and Systems Services Revenue
as a Percentage of Total Service Revenue

	Size classes by $1000s total operating revenue				
	less than 100	100-499	500-1999	more than 1999	all establishments
1972	25.85	18.23	20.64	16.46	18.73
1973	24.97	24.72	15.57	21.19	20.56
1974	-	-	-	-	24.00
1975	43.80	28.12	23.77	13.10	18.20
1976	46.37	27.74	24.48	12.60	18.19
1977	48.56	29.87	20.15	14.81	18.60
1978	*	31.24	25.50	*	21.09

*Data not reported to preserve confidentiality.

(e) Ability to survive and optimal size. In a market economy, those firms best suited to deal with market forces will survive and those ill suited will fall by the wayside. If firms of a certain size are better suited, they can in some sense be considered as optimal-size firms and will, as a class, gain an increasing market share over time. Firms in size classes which are particularly poorly suited for dealing with market forces will eventually decline in market share.[25]

Based on data for the years 1975 to 1978, during which the Statistics Canada criteria for inclusion in the survey appear to have

25. Studies implementing the survivor technique began with Stigler, 1958. See also Saving, 1961; L. Weiss, 1964; Weiss, 1965. For a thorough but overly critical discussion of the problems involved with the survivor technique, see Shepherd, 1967. One important caveat concerning survivor studies is that they indicate the staying power of firms in the face of all forces, including not only market but political forces.

remained unchanged, none of the size classes of computer services establishments declined in terms of the number of establishments listed in each size class (Table 1). There is an indication in the table, though, that the number of establishments increased more rapidly in the smallest size class than in the larger three size classes from 1975 to 1978. This fact by itself suggests that there are no economies of scale in the computer services industry and that the industry may even have some diseconomies of scale.

In the software and systems services segment of the computer services industry, Statistics Canada provides no data for the number of establishments over time, but it does provide sufficient data to calculate the share of market revenue accounted for by establishments in each size class (Table 5). While there are too few observations to perform statistical tests of significance and while there is some year-to-year variation in the market shares, it appears that establishments in the middle two size classes have been losing some of their market share to establishments in the smallest and largest size classes.

Table 5

Percentage of Market Revenue (i.e., Market Share) in Software and Systems Services Accounted for by Establishments in Each Size Class

	Size classes by $1000s total operating revenue			
	less than 100	100-499	500-1999	more than 1999
1972	5.81	14.03	32.15	48.01
1973	3.79	15.84	16.46	63.91
1975	5.52	21.64	28.08	44.76
1976	7.09	19.64	30.57	42.70
1977	6.71	19.73	21.81	51.75
1978	*	16.79	23.53	*

*Data not reported to preserve confidentiality.

Further confirmation of this finding is presented in Table 6, which shows the average market shares (times 100) for establishments

in each size class. This average has been declining dramatically for each size class because of the considerable amount of entry into the industry. However, as the last row of the table shows, it has been declining even more rapidly for establishments in the middle two size classes than for establishments in the largest and smallest size classes.

Table 6

100 times the Market Share of Each Size Class Divided by Number of Establishments per Size Class

	Size classes by $1000s total operating revenue				
	less than 100	100-499	500-1999	more than 1999	average for all establishments
1972	3.95	14.77	69.89	228.62	32.36
1973	2.67	14.14	35.02	213.03	30.21
1974	-	-	-	-	28.97
1975	3.83	13.53	44.57	149.20	25.19
1976	3.58	10.97	38.70	129.39	20.45
1977	2.73	9.18	24.78	110.11	16.78
1978	*	6.56	21.79	*	14.33
% decline from 1975-77	40%	47%	80%	35%	50%

*Data not reported to preserve confidentiality

What emerges from this discussion of the data presented in Tables 5 and 6 is that if establishments of some sizes are more nearly optimally suited for the Canadian market than others, they are either very small or very large, relatively speaking. This finding is not atypical for many survivor studies. It results in large measure from the heterogeneity of the product being sold by the industry. The smaller establishments are usually involved in specialized activities or are branch operations of large firms, while the large establishments generally offer a very broadly diversified range of products and services. The two stereotypical types of establishments can survive side

by side serving different subsectors of the industry. The results also are not inconsistent with Gibrat's Law of random growth in a competitive industry with easy entry.

Business Week also observed similar tendencies in the software industry and attributed them to technological growth and the increasing costs of custom and in-house software development (1980, p. 54). They point out that packaged programs have become increasingly attractive to potential customers over the past decade, in part because the packages have been refined and improved with experience and in part because it is often cheaper for customers to purchase a package, even if it isn't precisely what they want, than it is to have the software developed for them on a custom basis or in-house. As a result of economies of scale in the development of software packages and firmware and/or economies of large-scale marketing efforts, large firms are beginning to dominate this segment of the industry through internal software development and through the acquisition of smaller firms. Small firms continue to thrive, for the present, because they develop unique programs for certain market niches and can sometimes offer local customers more personal service. Because of this trend toward the growth of large firms surrounded by specialized smaller firms, the impact of additional legal protection for the intellectual property embedded in software on market concentration may become a more serious issue.

Although Statistics Canada data provide strong evidence that the Canadian software industry is competitive, the two aspects of its survey techniques mentioned at the outset -- first, that the data are collected for establishments, not firms, and second, that the data include all computer services establishments regardless of whether they produce any software -- present serious difficulties in using these data to draw this conclusion. To overcome these problems, a different set of data was collected which gives a slightly different perspective on the industry.

Canadian Datasystems surveys. An annual survey of firms in the broadly defined computer industry has been published by Canadian Datasystems since 1972. From 1972 to 1977, this survey identified those firms (not establishments) offering systems and programming analysis and packages. These features of the survey were extremely desirable in light of the deficiencies (for the purpose of this paper) in the Statistics Canada data. The survey does, however, have other weaknesses. It provides no data for three large firms in the industry: IBM, Control Data and Canadian General Electric. As well, its coverage appears to be incomplete. Many of the smaller firms may not have been included in the survey and no data are reported for some of the firms in some years. Finally, the survey provides data on the number of employees of each firm and the numbers of systems and programming employees but gives no data on total revenues or on the software and systems revenue of the firms.

Table 7

Number of Firms and Average Number of Employees per Firm

Year	Number of firms*	All employees	Number of firms*	Systems and programming employees
1972	70	43.91	65	16.52
1973	84	40.92	78	14.91
1974	130	46.00	125	16.40
1975	125	48.51	119	17.56
1976	166	43.80	158	13.96
1977	228	41.39	222	12.96
% change from 1972-77	226%	-5.7%	242%	-21.5%
% change from 1975-77	82%	-14.7%	87%	-26.2%

SOURCE: Canadian Datasystems

*"Number of firms" is the number for which data on the number of employees are available.

(a) Number and sizes of firms. Table 7 shows the number of firms listed in the survey as offering systems and programming services and for which employment data were published. As the Statistics Canada data showed, there has been phenomenal growth in the number of firms in the industry and the average size of the firms (measured in terms of numbers of employees) has declined somewhat from 1972 to 1977. Two sets of data are reported in Table 7 and later tables. The two columns on the left show the numbers of firms for each year and the average number of all employees per firm (for which data were published) for each year. The two columns on the right show these same figures for just systems and programming employees. In both cases, the average

number of employees per firm has declined over time but the average number of systems and programming employees has declined considerably more rapidly than the average number of all employees. The decline in the number of systems and programming employees relative to all employees is consistent with Frank's more qualitative and subjective statements that successful software houses are devoting more resources to marketing and less to software and systems development and maintenance.[26]

The survivor studies based on Statistics Canada data indicate that establishments in the middle size classes might be at a slight disadvantage relative to establishments in the largest and smallest size

---

26. Frank writes that:

> Based on a number of different organizations' experience [sic], the composite model of the software product firm is as follows:

Revenue	100%
Expenses	
Engineering	
Development	10%
Maintenance	10%
Marketing	50%
General Administration	10%
Total Costs	80%
Gross Profit	20%

> where development includes all ongoing enhancements and new product innovations, and maintenance supports existing products (Jan. 22, 1979, In Depth p. 16).

These figures are highly stylized, however, because the 1978 ADAPSO Industry Report indicates that, in 1977, the average gross profit rate on revenues for 60 respondents was only 11 per cent, not the 20 per cent figure put forward by Frank. Nevertheless, the point deserving emphasis here is that marketing is becoming preeminent in the computer software industry. Frank's assertion to this effect was confirmed in Business Week, September 1, 1980, p. 55.

classes. Data from the Canadian Datasystems surveys indicate a similar phenomenon for firms in the Canadian computer software industry. Tables 8 and 9 show the numbers of firms by size category. In Table 8, the size categories are based on numbers of all employees; in Table 9, the size categories are based on the number of systems and programming employees. The last row of each table shows the percentage change in the number of firms in each size category from 1972 to 1977.

Table 8

Number of Firms in Each Size Category

	Size of firm by total number of employees					
Year	0-10	11-30	31-50	61-100	more than 100	total
1972	19	20	17	9	5	70
1973	25	27	17	8	7	84
1974	44	40	18	13	15	130
1975	36	43	15	15	16	125
1976	54	51	23	20	18	166
1977	87	72	24	21	24	228
% change from 1972-77	358%	260%	41%	133%	380%	226%

SOURCE: Canadian Datasystems

Although there has been an increase in the number of firms in each size category, that increase was considerably smaller for the middle size classes than for the smallest and largest size classes (Table 8). This pattern does not carry over to Table 9, though, where it can be seen

that the increase in the number of firms with fewer systems and programming employees was greater. This finding provides further support for the hypothesis that marketing employees are becoming more important in the industry and it appears that this hypothesis has the most weight among the larger firms.

Table 9

Number of Firms in Each Size Category

Size of firm by number of systems and programming employees

Year	0-5	6-10	11-20	21-50	more than 50	total
1972	25	16	10	8	6	65
1973	33	15	15	10	5	78
1974	55	30	20	11	9	125
1975	53	23	21	11	11	119
1976	80	33	22	13	10	158
1977	121	42	28	17	14	222
% change from 1972-1977	384%	162%	180%	112%	133%	241%

SOURCE: Canadian Datasystems

(b) Turnover among computer software firms. The Canadian Datasystems survey can be used to gain a better perspective on "chaos" in the industry by examining entry and exit rates in each year (Table 10). From 1972 to 1977, there were 266 entrants into and 108 exits from the Canadian software industry. Again, since the survey coverage was incomplete, the high entry rates may in part reflect increased coverage by the survey over time. As shown in the last row of Table 10, the entry rate averaged 36.8 per cent and the exit rate averaged 18.9 per cent. The last column of Table 10 shows the number of firms entering the industry in one year but exiting within the next year. On the average, 26.7 per cent of all entrants exited from the industry

Table 10

Turnover in the Canadian Software Industry

	Entrants		Exits		Exits After One Year	
Year	number	as % of total	number	as % of total in previous year	number	as % of entrants in previous year
1973	31	36.9	11	15.7	-	-
1974	67	51.6	19	22.6	10	32.2
1975	34	27.2	29	22.3	22	32.9
1976	58	34.9	23	18.4	6	17.6
1977	76	33.3	26	15.7	14	24.2
Total/ average	266	36.8	108	18.9	52	26.7

SOURCE: Canadian Datasystems

within one year. Although both the entry and exit rates are somewhat higher than those in other competitive industries, the exit rate of new entrants is generally lower than that experienced in other competitive industries.[27] These comparisons suggest that the computer software industry is young and growing rapidly. Entry and exit rates are both high due to the growth and technological progress in the industry, yet entrants have a reasonably good chance of survival (relative to entrants in other industries) because of the growth of the industry.

Tables 11 through 16 break down the entry, exit and exit-after-one-year rates by size categories. The figures presented in these tables provide additional support for the conclusion tentatively reached earlier in the paper that firms in the middle size categories appear to be at a slight disadvantage relative to those in the smallest and largest size categories. Using total employment to measure size (Tables 11 and 13), entry rates were greater than exit rates for firms with less than 10 or more than 60 employees, but exit rates were greater than entry rates for firms with between 11 and 60 employees. These divergences in entry and exit rates presumably would not have occurred unless medium-sized firms were at some disadvantage relative to the others. Similar though less clear results appear in Tables 12 and 14, where the size of firms is measured in terms of the number of systems and programming employees. Regardless of the measure of firm size, the average size of entering firms was slightly smaller than the average size of exiting firms.

Even though small firms seem to have a slight advantage over medium-sized firms in general, they might also have more difficulty weathering the storms of their first year of operation (Tables 15 and 16). While the average size of all exits from the industry was 24.3 employees, the average size of entrants exiting after one year was only 18.1 employees. In part, this result obtains because of the high entry rate of small firms. Normalizing for entry rates by calculating exits after one year as a percentage of entrants in the previous year (the last rows of Tables 15 and 16), the entering firms in the smallest and largest size classes again appear to have had a slight advantage over entering firms in the middle size classes when firm size is measured using total employment. Using systems and programming employment, however, this pattern is not repeated, perhaps due in part to the small number of observations reported in the various cells of Table 16.

---

27. Table C-1, Appendix C, shows entry, exit, and exit-after-one-year rates for a number of low overhead, easy entry industries. The data are taken from a preliminary version of a study on municipal transportation regulation for the Ontario Economic Council by John Palmer and Kenneth Engelhart.

Table 11

Entry by Size of Firm
(measured as total number of employees)

	Size classes						
Year	0-10	11-30	31-60	61-100	more than 100	all firms	Average number of employees per entrant
1973	15	8	3	0	1	27	17.3
1974	29	19	9	6	1	64	22.8
1975	19	9	3	2	1	34	31.9
1976	27	14	7	5	3	56	25.6
1977	43	17	8	3	2	73	20.0
Totals	133	67	30	16	8	254	23.2
As % of all entrants	52.4%	26.4%	11.8%	6.3%	3.2%	100%	

SOURCE: Canadian Datasystems

Table 12

Entry by Size of Firm
(measured as number of systems and programming employees)

	Size classes						
Year	0-5	6-10	11-20	21-50	50	all firms	Average size of entrants
1973	18	5	2	1	0	26	6.46
1974	32	15	11	3	1	62	8.60
1975	24	6	1	1	1	33	11.48
1976	35	10	5	3	0	53	6.75
1977	52	11	4	3	3	73	7.19
Totals	161	47	23	11	5	247	7.95
As % of total entry	65.2%	19.0%	9.3%	4.4%	2.0%	100%	

SOURCE: Canadian Datasystems

Table 13

Exits by Size of Firm
(measured as total number of employees)

	Size classes						
Year	0-10	11-30	31-60	61-100	more than 100	total	Average size of exiting firm
1973	4	4	2	1	0	11	23.6
1974	5	6	5	1	1	18	33.1
1975	15	11	3	0	0	29	13.8
1976	10	6	2	2	2	22	35.3
1977	14	6	4	2	0	26	21.0
Total	48	33	16	6	3	106	24.3
As % of all exits	45.3%	31.1%	15.1%	5.7%	2.8%	100%	

SOURCE: Canadian Datasystems

Table 14

Exits by Size of Firm
(measured as number of systems and programming employees)

	Size classes						
Year	0-5	6-10	11-20	21-50	more than 50	total	Average size of exiting firms
1973	5	3	1	1	0	10	9.50
1974	6	6	2	2	0	16	11.69
1975	20	6	0	1	0	27	5.56
1976	12	1	5	2	2	22	16.91
1977	17	3	3	1	0	24	6.42
Total	60	19	11	7	2	99	9.67
As % of all exits	60.6%	19.2%	11.1%	7.1%	2.0%	100%	

SOURCE: Canadian Datasystems

Table 15

Exits after One Year by Size Class
(measured as total number of employees)

	Size classes						
Year of exit	0-10	11-30	31-60	61-100	more than 100	total	Average size
1974	3	3	1	0	1	8	29.5
1975	11	9	2	0	0	22	13.4
1976	5	1	0	0	0	6	7.2
1977	5	5	3	1	0	14	23.3
Total	24	18	6	1	1	50	18.1
As % of all exits after one year	48%	36%	12%	2%	2%	100%	
As % of entrants in previous year	26.7%	36.0%	27.2%	7.7%	16.7%		

SOURCE: Canadian Datasystems

Table 16

Exits after One Year by Size Class
(measured as number of systems and programming employees)

	Size classes						
Year of exit	0-5	6-10	11-20	21-50	more than 50	total	Average size
1974	3	2	1	1	0	7	11.71
1975	15	5	0	0	0	20	4.20
1976	6	0	0	0	0	6	2.66
1977	8	1	3	1	0	13	8.77
Total	32	8	4	2	0	46	6.43
As % of all exits after one year	69.6%	17.4%	8.7%	4.3%	0	100%	
As % of entrants in previous year	29.4%	22.2%	21.7%	25.0%	0		

SOURCE: Canadian Datasystems

The past and the future. Two occurrences related to antitrust suits involving IBM in the United States may have contributed to the comparative youth and growth of the software industry. In the first of these, IBM agreed in a consent decree to operate its Service Bureau Corporation as an arm's-length subsidiary in the United States, charging it the same prices for similar products and services as were charged to other customers.[28] This agreement opened the way for the growth of independent computer service bureaux in the 1960s. In the second occurrence, IBM was charged in 1969 with monopolization, in part because it

28. U.S. v. IBM (1956 CCH Trade Cases, 68, 245 [S.D.N.Y. 1956] amended, Civil Case Number 72-344 [S.D.N.Y. 1963 and 1970]). As part of an out of court settlement of a private antitrust suit, IBM sold Service Bureau Corp. to Control Data in 1973 for its book value. Although these actions took place in the United States, they opened the U.S. market to both Canadian and U.S. software suppliers.

marketed its hardware and software together as a package deal for package prices.[29] Needless to say, independent software developers found marketing their products difficult when users in essence paid no incremental fee for IBM's software. Shortly after the charges were filed, however, IBM "unbundled" its products, setting separate prices for hardware and software. Regardless of IBM's reasons for its unbundling decision, independent software houses consequently found entry and growth in the software market considerably easier during the 1970s. By 1978, according to a list published by International Computer Programs, Inc., there were more than 1000 companies supplying more than 6000 software products: 4000 for mainframe use and 2000 for minicomputer use. Of these, IBM had the following market shares:

	Mainframe
System	21%
Application	8%
	Minicomputer
System	13%
Application	10%

SOURCE: Frank, 1979, In Depth p. 4

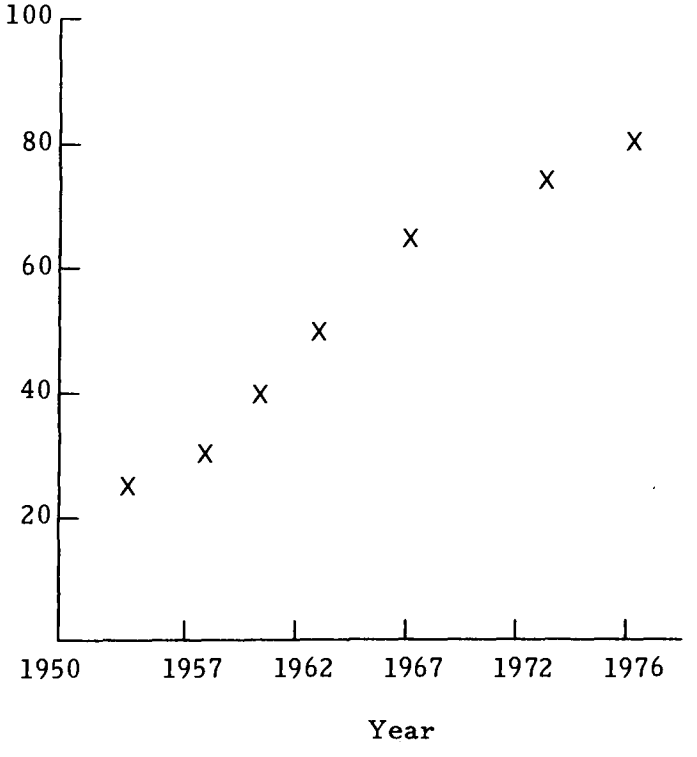
Another reason for the growth and deconcentration of the software industry has been the increasingly important role of software vis-à-vis hardware in users' expenses. By the late 1970s, more than 80 per cent of users' computing budgets went for software, as compared with less than 40 per cent in the late 1950s. This trend is depicted in Figure 1.

Despite the increasing importance of software expenditures in user budgets, purchased software (as opposed to custom-developed or in-house developed software) accounted for a rather small portion of the software industry in the 1970s. Different industry estimates placed the in-house software development portion of the market in the mid-1970s at

29. U.S. v. IBM (Civil Case Number 69-200 [S.D.N.Y. 1969]). This case is still at trial. The unbundling took place in Canada as well.

Figure 1

Software Cost as a Percentage of Total System Cost



SOURCE: Electronics Systems Division, Air Force Systems Command, Support of Air Force Automatic Data Processing Requirements Through the 1980's 2-121 (1974) (Vol. 3-Technology) (Hamscom Field) (A.D. Little Report)

around 83 per cent to 85 per cent (Frank; Jan. 15, 1979; In Depth p. 5). They also placed outside purchases of software at between only one and two per cent of total data processing expenditures (Frank; Jan. 8, 15, 1979; In Depth p. 11, 5). With all software accounting for perhaps 80 per cent of data processing expenditures, software purchased outside accounted for a very small portion of the market, with custom development accounting for most of the remainder of the market. There is evidence, however, that the purchased software share of software expenses will be growing in the future. As already cited, Business Week expects that package software will become increasingly important in the software market, eventually dominating other software if it doesn't already. A case study in the Harvard Business Review dis-

cussed a firm which purchased software, essentially off the shelf, from the supplier for $100,000. The authors estimated that equivalent software would have cost the firm over $1 million and taken between two and three years to develop in-house (McFadden and Suver, 1978). While this example may be fairly dramatic, interviews with other users and recent business publications have revealed that users are becoming increasingly reluctant to develop their own software when much has been developed already and is available from outside sources already debugged and costing considerably less.

Perhaps the most important development in the software industry will be the integration of software with hardware in what has come to be known as firmware: microprocessing silicon chips or cartridge tapes with software programmed onto them. Firmware is often plug-compatible with hardware and, because the physical units are inexpensive to mass produce once they have been developed, they offer a new way to market on a major scale software which is of value to many users. The actual process of programming microprocessing units does not necessarily require large-scale investments; small firms can purchase the units and do the programming quite easily or can contract to have the program fed into a read-only-memory in the units. However, markets for general use programs will evolve as firmware becomes less expensive than traditional software and the nature of these markets will be such that firms will have to devote considerable resources to marketing their products, once developed. In other words, the emergence and growth of firmware technology could lead to increased concentration in at least some portions of the software industry, depending on both the size of the marketing effort needed and the availability of financial capital for small firms to market their products on a sufficiently large scale.

Relevance for intellectual property protection of computer software. It has been argued that, because of the importance that marketing has taken on for packaged computer software, there will be a trend toward increased concentration in the software industry, both as it has been known traditionally and in the firmware sector of the industry. Braunstein et al. (1977) agree with this argument in their report for CONTU. Yet they also point out that trade secret protection of software also confers some advantages on larger firms offering a full line of software. When computer programs are protected by trade secrecy, potential clients are less likely to be aware of precisely what programs are available and from whom. Customers have high shopping costs and the software developers have high marketing costs as they attempt to inform the market as to what is available while at the same time protecting their trade secrets. Braunstein et al. argue that providing copyright protection of computer software will reduce this trend toward increasing concentration in the industry by reducing marketing and information costs. We are less sanguine. It is not at all clear (and no evidence could be found on this point) whether small or large firms are more likely to appropriate the software developed by others. If small

firms are the beneficiaries of a lack of copyright protection for software, on balance appropriating more from large firms than large firms appropriate from them, then providing copyright protection for computer software could easily lead to increasing, not decreasing, concentration in the industry. And, to the extent that the large firms have more experienced legal staff on retainer or in-house, they may have potential cost benefits over smaller firms if copyright infringement litigation is threatened.[30]

Chapter III returns to this discussion of the effects of copyright protection on the structure of the computer software industry and offers policy analysis and recommendations. Before beginning that discussion, though, it is important to understand the present legal environment. The next chapter discusses the economic rationales for intellectual property protection, followed by a discussion of intellectual property law as it presently applies to computer software.

---

30. Smaller firms would be at a disadvantage with respect to legal costs if legal services are not perfectly divisible into small units for smaller, infrequent clients (i.e., if there are some economies of scale in the production of legal services in the lower ranges of output).

Chapter II

INTELLECTUAL PROPERTY PROTECTION

Introduction and Rationales

Some form of legal protection for the fruits of intellectual effort has existed for centuries. Early monarchs and rulers granted exclusive production rights to loyal subjects as a reward for faithful service and as a method of facilitating censorship. Often these exclusive monopoly rights were little more than the result of rent-seeking behaviour on the part of the subjects and an attempt by the rulers to share the potential crown-created monopoly rents. Only through historical experimentation with different forms of monopoly grants, and especially during periods when they were not enforced, did people begin to recognize more clearly the public-good nature of much intellectual effort. Certainly the invention of the printing press helped to speed this realization.[1]

Public goods and intellectual effort. After an economic agent engages in commercially valuable intellectual effort, and once it is marketed, the fruits of the effort are often easily appropriable by others in the absence of intellectual property protection. For example, a new product can be purchased or even simply examined by another producer who, in turn, can then produce the new product without bearing the investment costs of invention, development and initial marketing. Similarly, once an artistic or literary work is created, others can appropriate the creation and reproduce it without bearing the initial creation and marketing costs. In each case, the innovation or creation is the embodiment of intellectual effort, yet once the output of the intellectual effort becomes known, the initiator of the effort has no claim to it or to its rewards. Because intellectual output becomes a free good and imitative entrepreneurs will tend to copy the more valuable intellectual output, those who have the potential of generating the output will have less incentive to do so. Fewer resources will be devoted to intellectual effort if its output carries no exclusive right of remuneration but is instead produced at a competitive rate of output with greater output and lower prices. In other words, a form of static allocative efficiency will obtain, perhaps at the expense of future output of intellectual effort (i.e., perhaps at the expense of dynamic efficiency).

Reducing appropriability. One of the major causes of the public good nature of the output of intellectual effort is the ease of appropriability. If the creator of the output could market it to all potential cus-

---

1. For complete treatises on the history of intellectual property protection, see Patterson, 1968; Wincor, 1967; Ladas, 1930; Fox, 1969; Machlup, 1958. Arthur Plant has attacked the traditional rationales for intellectual property protection in two 1934 Economica articles. See also Hurt and Schuchman, 1966. One of the best discussions of the economic rationales for intellectual property protection appears in Hindley, 1971, a study which is unfortunately relatively obscure.

tomers before someone else could appropriate it, there would be no incentive problem; imitators would have no incentive for appropriation and creators would receive all of the potential rewards for their efforts. Creators would receive monopoly rents (if merited by the market demand for their output) and would have an incentive to be creative with the hopes of receiving large monopoly rents. Each specific creation would be produced at less than competitive rates of output and greater than competitive prices. Although such a monopoly would very likely be quite limited in scope, it would nevertheless give rise to static inefficiency with allocative distortions in the production of each specific work. It would also be more likely to engender dynamic efficiency by encouraging the devotion of more resources to creative activity. In most cases, however, the creator would be unable to market his output to all potential customers before an imitator could appropriate it. In the absence of explicit legal intellectual property protection, there are basically five methods extant for limiting appropriability or the effects of appropriability by imitators.

(a) Rapid marketing. The creator of a work can market his output quickly to a large proportion of purchasers before imitators have a chance to appropriate the work and market it themselves. The feasibility of this tactic depends on a number of conditions:

(1) The product must be capable of being marketed quickly. Otherwise, imitators will be able to appropriate it before the creator has reached much of the market.

(2) Potential customers must be assured that appropriators will not soon be able to market the product at a lower price so that they do not decide to delay their purchase.

(3) Financial capital market imperfections must not be such that they significantly raise marketing costs.

(4) Market durability of the product must be less than the appropriation lag. For example, if the product has a market life of 20 years but the appropriation lag is 30 years, the tactic can work. If, however, the market life is 16 months but the appropriation lag is only 16 hours, the tactic would not be very effective.[2]

---

2. Frank reports on a study by Kendall which reviewed life cycle statistics for a population of 5,328 application programs developed within several IBM organizations and found the average life span of a program to be about 16 months (January 8, 1979, In Depth p. 10).

If the tactic is completely successful, the creator will receive the same revenues as would be generated under an intellectual property protection statute. His costs would differ by the difference between the costs of more rapid marketing and the costs of pursuing legal protection under the statute.

(b) Price discrimination. If the creator can identify which customers will sell copies of the output and how many each will sell, then, in the absence of legislation prohibiting price discrimination, the creator can charge each customer a price which captures the full monopoly rent according to the number of copies made. In cases for which perfect price discrimination is not economically feasible due to high information costs, second degree price discrimination might be possible based actuarially on the likelihood that different classes of customers would produce some arbitrarily large quantity of copies.[3]

(c) Custom production. The creator of a work can, to some extent, protect the value of the output of his intellectual effort by producing output which is of greater value for one specific customer than it would be for others. Consider, for example, a suit tailor-made for individual A. Individual B might have an incentive to appropriate (i.e., steal) A's suit, either to resell it to someone A's size or to have it remade for himself. Because the suit does not have a generality about it, because few others can use it without additional expense (even the expense of finding another buyer), the expected net benefits of appropriation are less than they would be for products with greater generality. The analogy of a tailor-made suit can be generalized to many other creative works. A machine designed to perform specific tasks for one firm might have features valued by other firms and yet require alterations in other features in order for appropriation to be feasible. Similarly, a computer software package designed for one client might require substantial adaptation to allow it to be used on another client's computer or for yet another client's slightly different objectives. In some cases, the alteration and adaptation costs may be very small and custom production would therefore be of marginal value in protecting the creator's intellectual property. In other cases, however, the uniqueness of the cus-

---

The different lags enabled an imitator to market a computer chess game approximately one year after it was marketed by its originator. Data Cash Systems, Inc. v. JS & A Group Inc., Docket No. 79 591 (N.D. Ill., Sept. 26, 1979).

3. For a discussion of this proposition, see Benjamin and Kormendi, 1974. Their treatment was improved, extended and applied to photocopying by S.J. Liebowitz, 1980(b).

The discussion here is not intended to imply that the only reason firms might practice price discrimination is to appropriate revenues from imitators.

tom product could mitigate the detrimental effects which imitation might have on the creator's revenues.4

(d) Technical protection. Suppose that Chester Carlson had been able to produce his first selenium-coated plate for photocopying machines so that, when it was opened, a chemical reaction would take place, destroying the contents beyond recognition. He would not have needed patent protection. A more relevant example of technical protection is the use of software locks and a series of codes by computer service firms to guard against the invasion of privacy. Such devices and schemes are also used to protect the intellectual property embodied in software. Another example comes from the realm of telecommunications, where scrambling and unscrambling devices are used to limit the appropriation of pay television broadcasts. That technical protection is imperfect is evident from two recent experiences in the United States. One involved a high school student breaking the code of a computer service operation and gaining access to confidential data. The other involved the sale of unscramblers in the Los Angeles area to non-subscribers, enabling them to view ON-TV's pay television programs. Nevertheless, to the extent that technical protection increases the costs of appropriation, it reduces the anticipated net benefits accruing to imitators and increases the returns to creators by discouraging imitation.

(e) Exclusive contract protection. As the common law has evolved, a fairly sophisticated system has developed allowing creators to protect their works via equity or via express or implied contracts. The creator must generally concern himself or herself with two groups of parties: employees and licencees. Employees can appropriate the output of the creator's intellectual effort and sell the output to another employer by changing jobs. Similarly, a customer can purchase the output from the creator, often through a licensing arrangement, and then resell or relicense it to other potential customers. In both cases, the creator can protect against appropriation by using the law of contracts and trade secrecy to restrict the activities of the contracting parties.

Exclusive contract protection works best in situations involving products which would be marketed with explicit contractual arrangements anyway. In these situations, the addition of exclusivity and trade secret clauses contributes little to the contracting costs for the parties. This type of protection does not, however, lend itself to easy implementation in situations involving products which would generally be

---

4. Of course, in these cases appropriation would be less likely to be profitable for an imitator. It is questionable to what extent custom production is a viable tactic to pursue to protect intellectual property. It seems more reasonable to view custom production as a technological and market phenomenon which is not undertaken in and of itself for the sole purpose of increasing appropriation costs.

mass-marketed without contract negotiations for each transaction. In these situations, the product is usually mass-marketed in part to reduce the transactions costs for each sale. Requiring that each sale be accompanied by a contract spelling out non-disclosure agreements would increase the transactions costs and reduce the marketability of the product. In addition, such contracts would be extremely difficult to enforce due to difficulties in detection and problems arising from gifts and other non-market or after-market exchanges.

Contractual protection is common in the computer software industry. IBM has an 11-page contract which it considers standard for its customers. And, as Morgan says:

> Despite the thoroughly unsatisfactory nature of copyright and patent law as regards software, some software houses try hard to secure whatever may be secured and it will at least be possible to apply copyright to the documentation (manuals, etc.):
>
> > Patent copyright and other industrial property rights in the Package and its programs and any associated documentation shall be vested in the Licensor who reserves the right to sell the Package to any other party or parties.
>
> But the same contract recognizes that this is inadequate, for elsewhere it says:
>
> > The Licensee hereby agrees:
> > (i) that the Package is the sole property of the Licensor and that the Licensee will take all reasonable precautions to maintain the confidentiality of the Package its programs and documentation;
> > (ii) that it will not assign transfer mortgage charge pledge or sublet any of its rights or obligations under this Agreement;
> > (iii) to make no copies of or duplicate the Package or any part or parts thereof by any means or for any purpose whatever (except as may be necessary for normal security storage) without the prior consent in writing of the Licensor;
> > (iv) to use the Package solely at the installation described in the Schedule hereto;
> > (v) to instruct all its staff from time to time having access to the Package not to copy or duplicate the Package or any part or parts thereof or to make any disclosure relating thereto to any third party;

(vi) to effect and maintain adequate security measures to safeguard the Package from theft or access by any person other than employees of the Licensee in the normal course of their employment;
(vii) in the event that any of the programs comprising the Package or any part or parts of the associated documentation should come into the hands of a third party through the Licensee or any employee or former employee of the Licensee, the Licensee shall forthwith pay to the Licensor the price for the entire Package ruling for the time being as would be charged such third party for a Licence to use the Package.

This is about as complete as human ingenuity can make it. Notice that the mention of "former employee" in clause (vii) stops up a loophole which is not otherwise dealt with, and effectively lays upon the Licensee the duty to add a clause as to the security of this package to his contracts of employment with his employees. Yet I would say the licensor is fully justified in this stricture, for I have known cases where a disreputable rival service bureau has recruited staff from a licensee expressly to learn about a licensor's package.

Sometimes the licensor is happy for the licensee to pass the package to third parties, provided he does so as the licensor's agent:

The Licensee may act as agent for the Licensor in the grant of other Licences for the Package:
(i) The Licensee shall have no power express or implied to bind the Licensor in any way whatever;
(ii) If such licence to a third party is granted through the efforts of the Licensee the Licensor shall be entitled to the full price for the time being of the Package from such third party and in the event of any default in such payment the Licensee hereby agrees to indemnify the Licensor for the full price for the time being of the Package;
(iii) If such other licence is granted to a third party for a fee wholly through the efforts of the Licensee then upon the Licensor receiving the full price for the time being of the Pack-

> age (whether from the third party or from the Licensee) the Licensee shall be paid by the Licensor a commission of x% of the price for the time being of the Package;
> (iv) The Package shall be supplied only by the Licensor and upon the terms and conditions contained herein and the Licensee has no authority express or implied to vary add alter or amend such terms and conditions in any way whatsoever.

Finally, the licensee should satisfy himself that the licensor is, in fact, the originator and owner of all parts of the package being sold, and it is not unreasonable for him to demand an indemnity in case of any future dispute:

> The Licensor shall indemnify the Licensee against all claims demands costs charges and expenses arising from or incurred by any infringement of copyright patent or other title in respect of the Package or any part thereof provided that such infringement is not caused or contributed to by any act or acts of the Licensee other than the use of the Package in accordance with the provisions of this Agreement. (Morgan, 1979, p. 66)

Interestingly, even though such contracts have come to take on a fairly standard form, in a surprisingly large number of cases, usually involving small, first-time users, contracts are not drawn up.[5]

Intellectual property protection in the law. Each of the methods for limiting uncompensated appropriation imposes costs on the creator and hence on society. The creator must devote additional resources to marketing speed and skills, to technological research for protection, to increased secrecy or to legal expenses. Additional social costs are created (except possibly in the case of rapid marketing) with most of the strategies because they require devoting resources to the protection of secrecy and ideas are not as freely disseminated as they might be under some alternative mechanism for intellectual property protection. The argument raised earlier in this chapter -- that unprotected works would be produced at a competitive rate of output and at a competitive price -- seems highly questionable. The lack of intellectual property

---

5. See Douglas, 1973 and the section on trade secrets and computer programs later in this chapter. Jack Roberts indicated to the authors that equity often protects a party even in the absence of a contract.

protection under the law would both impose additional costs on producers and encourage them to restrict their output in order to control their trade secrets. As a consequence, it is not clear that the lack of intellectual property law engenders static efficiency, much less dynamic efficiency. In making this point in his discussion of crime, Posner says: "Observe the consequences if theft were freely permitted. Property owners would spend a great deal of money on devices for the protection of property and would substitute otherwise less valuable goods that happened to be less easy to steal," and adds, as a footnote, "Do you see a parallel to the use of trade secrets as a method of securing property rights when patent protection is unavailable?" (Posner, 1972, p. 68).

Almost as if in answer to the question posed by Posner, Braunstein et al. write:

> [T]rade secret protection biases the quality of new output in the direction of unnecessary complexity. It also channels the innovative efforts into those forms of software which are not particularly well-suited to general use. In other words trade secret protection is particularly detrimental to private provision of non-specialized, general interest types of software. Copyright protection diminishes that type of distortion....
>
> Abstracting from a product's characteristics, we may inquire about the supply of products under various forms of protection. The general result which emerges from theoretical discussions is that irrespective of the kind of protection, competition will tend to overprovide inventions and innovations with high private value, and simultaneously underprovide those with high social value. These are two opposing forces, one of which tends to increase the supply of inventions over the socially optimal level, and another tendency which decreases the amount. Which effect will outweigh the other depends on the specific situation. There will be an increased supply of inventions and innovations which bring a private advantage to the supplier, possibly at the expense of the rest of society. On the other hand, inventions which are useful to society as a whole, but difficult to sell and therefore of relatively small value to the producer, will be supplied in a less than socially optimal amount. For example, programs which imitate or duplicate existing programs and therefore add very little to social welfare, but which enable the producing firm to capture a portion of the corresponding market and enter a lucrative business, will tend to be overproduced. But programs of a highly innovative nature, which could be beneficial for a wide variety of users for a long time period, but which

> it may be more difficult to market, will tend to be underproduced. The net effect can be an increase or decrease in innovative activity from the socially optimal level, but it is clear that with competition the wrong kind of innovation will be forthcoming. It is also clear that with competition a larger amount of resources will be devoted to the production of any given amount of innovation than is socially necessary.
>
> Under trade secret protection, the dangers of overprovision are magnified because there is lack of information as to the existing products. Overprovision in the regime will take the form of too many identical products being developed by independent producers. As we have seen, however, the variety may also be affected adversely. (Braunstein et al., 1977, pp. III-2, 12, 13)[6]

Because society desires both static and dynamic efficiency and because it is likely that these goals will not be approached in the absence of a more formal and legal recognition of intellectual property rights, most governments have enacted, amended and extended such legislation in the belief that the efficiency gains will more than outweigh the efficiency losses.[7]

(a) Patent protection. Patent law is designed to protect ideas in the physical arts and sciences. The ideas may be embodied in either a machine or a process. The term for patent protection is 17 years in both Canada and the United States.

---

6. See also Hirshleifer, 1971.

7. Some academics view the use of trade secrecy almost as a necessary evil and encourage the extension of copyright and patent protection in the law:

> Given the defects of trade secrets, it seems that there is something to be gained in the direction of refining the criteria for drafting and enforcing patent [and copyright] claims so that the use of the trade-secret option may be reduced....[T]rade secrecy as an option should be left open if only for its inseparable relation with the free enterprise system as a whole; on the other hand, alternative systems such as patent or copyright should be made more accessible, to reduce the optional use of trade secrecy. (Cheung, 1979, p. 25)

As will be seen in Chapter III, much trade secrecy is now practised in the computer software industry to protect ideas which could not be protected in any other way, even if computer programs could be patented or copyrighted.

(b) Copyright protection. Copyright law protects the expression of ideas in artistic or literary form. The term for copyright protection in Canada varies with the type of work, but it is usually at least 50 years and often is the life of the author plus 50 years.

(c) Trade secret protection. Trade secret protection is employed most often in situations involving ideas or expressions which cannot be protected by other laws. Classic examples of trade secrets include Stradivari's processing technique for wood used in violins and, more recently, the secret formulae for Coca-Cola or Kentucky Fried Chicken. Primary requirements for a trade secret to be recognized by the courts include evidence that it is (was) indeed a secret and that reasonable steps were taken to safeguard the secret. Trade secret protection can cost very little or a great deal, depending on the steps taken to insulate the secret, and it can last indefinitely.

(d) Industrial design protection. Industrial designs are typically ornamental or aesthetic, as opposed to functional designs for articles of commerce. They often are associated with particular brand names and are protected in part to facilitate the information flows associated with brand names. The term of protection for industrial designs is only five years, renewable for an additional five years. Because computer software is not primarily ornamental or aesthetic, it is an unlikely candidate for industrial design protection even though computer programs may be used to generate industrial designs.

(e) Trademarks. Trademark protection is used to protect the brand image of a particular product and thus offers some modicum of protection to consumers who come to rely on historic quality levels associated with different brands. Trademark protection is renewable indefinitely for 15-year terms. It has no special relevance for the protection of computer software beyond its general function of facilitating information flows concerning specific products.

Each of these five types of intellectual property protection is shown in Table 17 along with its individual characteristics. Against this general background of the rationales for and alternatives to intellectual property protection in law, the remainder of this chapter is devoted to a more detailed discussion of patent, copyright, and trade secret protection of computer software.

Patentability of Computer Programs

Jurisprudential background of patents. An invention, to be patentable, must satisfy a two-step investigation. First, the subject matter of the invention must fall within a permissible category. Permissible subject matter of a patent is restricted by statute to inventions or discoveries embodied in "any new and useful process, machine manufacture, or

Table 17

Intellectual Property
Comparison Chart

	General Subject Matter	Nature of Protection	Term of Protection	Qualifications for Protection	Compulsory Licensing	Cost	
						Time	Dollars
Patents	Ideas in the physical arts and sciences (i.e., new and useful art, process, machine, manufacture or composition of matter, or new and useful improvement therein).	1) Statutory protection 2) Limited monopoly giving inventor exclusive right to make, use or sell invention 3) Right against the world, not just copiers 4) Geographically limited to Canada	17 years	Idea must: 1) have practical utility; 2) be novel; 3) be unobvious to person of ordinary skill in the art; 4) not have been known or used anywhere in the world prior to date of invention; 5) not have been described in any patent of printed publication anywhere in world more than two years prior to date of application for patent and; 6) not have been in public use or on sale in Canada for more than two years prior to date of application for patent.	1) After three years from date of grant of patent if patent rights have been abused 2) Limited compulsory licences, with no waiting period, for food and drug patents	Two years (average) between application and issue of patent (examination procedure).	$1,000.00 (average)
Industrial designs	Shape or configuration (i.e., ornamentation) to be applied to articles of commerce that are to be multiplied by industrial process.	1) Statutory protection 2) Limited monopoly giving design registrant exclusive right: a) to apply design to articles of manufacture for purposes of sale; and b) to publish, sell or expose for sale or use an article to which the design is applied 3) Right against copier only 4) Geographically limited to Canada	5 years, subject to renewal for additional 5 years	Design must: 1) be new; 2) be orginal; and 3) not have been published in Canada more than one year prior to application for registration.	No	One year, perhaps less (examination procedure).	$500.00 (approximately)
Trade secrets	Secret machines, processes, formulas, industrial know-how	1) Common law, not statutory protection 2) Right to prevent others from making, using or disclosing 3) Right against wrongful appropriation of trade secret, not against world 4) No geographical limit	Protected until "secret" becomes common knowledge in industry.	Must be kept secret or disclosed solely under injunction of secrecy. (In certain circumstances, complete secrecy may not be required).	No	Variable	Variable

(cont'd)

Table 17 (cont'd)

Trade-marks	Marks or shapes which distinguish one company's wares or services from another's.	1) Common law and statutory protection 2) Common law protection arises from longstanding use in trade; limited to geographical area in which used; action is for "passing off" 3) Statutory protection confers exclusive right to use the mark throughout Canada, subject to good faith use by others prior to registration 4) Registration under statute condition precedent to infringement action under Trademark Act; protects against use of marks "confusingly similar" to registered mark.	No maximum. Statutory right subject to renewal every 15 years.	For statutory protection, trademark or trade name must be: 1) used (or proposed to be used within 6 months) to distinguish wares or services; 2) not confusing with any other trademark or trade name; and 3) not descriptive or deceptively misdescriptive of wares or services to which it is applied	No	Statutory protection: Less than one year on average (preliminary review procedure). Common law protection: At least 3-5 years (average) use in trade to develop common law right.	Statutory Protection: $400.00 (average) Common law protection: none
Copy-rights	Books, graphic arts, writings, sculpture, paintings, engravings, photographs, architectural works, lectures, compilations, abridgement, dance choreography, motion picture films, video tapes, phonograph records	1) Statutory 2) Right to prevent production or reproduction of works already published or works already composed but not yet published 3) No protection against similar works developed wholly independently of copyrighted work 4) No protection for ideas but for materials which embody the ideas 5) Protection extended beyond borders of Canada by international convention	1) Life of author plus 50 years for most works. 2) Phonograph recordings and photographs protected for 50 years from time original master or negative is made. 3) Crown copyright in work done under government direction runs for 50 years from date of first publication.	Work must: 1) be original; and 2) include some identifiable expense, labour, skill, judgement or imagination.	Yes, if owner of copyright is determined to be withholding work from public. Two cents per playing surface for sound recordings of musical works.	None. (Registration unnecessary to protection. Affixation of appropriate symbol and name sufficient to ensure international protection).	None

SOURCE: Adapted from notes by Jack Roberts for his course on intellectual property law at the University of Western Ontario. The authors are grateful for his permission to reproduce the table here.

composition of matter, or any new and useful improvement thereof."[8] Secondly, the invention must pass three separate tests: it must be useful (utility), it must be novel and it must be unobvious.[9]

The test of utility is satisfied if the invention can perform one of the beneficial functions that the inventor claims it can. Novelty is a more difficult test to pass. A patent claim lacks novelty if the elements of the alleged invention are present in a single prior art, technological structure or device which can perform substantially the same function in substantially the same manner. The requirement of novelty applies only to the combination of the constituent elements of the invention; there need not be novelty in its function.[10] Even if all of the constituent elements are disclosed in prior art, an invention may still possess novelty if it has brought together those elements in an unanticipated manner or unobvious manner.[11] For the third test to be met, the patent claim must disclose an inventive advance in the art. An invention may be simultaneously novel and obvious. For example, a coffee cup with four handles may be novel in the sense that no one has previously invented a cup of that particular type but it would be unlikely to pass the nonobviousness test. The test of inventiveness is not met if the association of the constituent parts of the patent claim would be obvious, given the existing state of the art, to a person skilled in that art. If an invention falls within the scope of permissible subject matter and satisfies the conditions of utility, novelty and unobviousness, the inventor will be granted patent protection in the form

---

8. "Whoever invents or discovers any new and useful process, machine, manufacture, or composition of matter, or any new and useful improvement thereof, may obtain a patent therefore..." (35 U.S.C. S. 101 [1976]). The Canadian legislation defines an invention in very similar terms as "any new and useful art, process, machine, manufacture, or composition of matter, or any new and useful improvement in any art, process, machine, manufacture or composition of matter;" (Patent Act, R.S.C., c. 203 s. 1 as amended). The addition of the word "art" in the Canadian definition has no significance for program patentability.

9. This follows from the statutory requirements of "new," "useful" and "invention."

10. The mere fact that there exists, in the prior art, a machine to perform a given task does not preclude an inventor of a new machine to perform that same task from seeking a patent. The classic example is a better mousetrap.

11. An example is a chemical process in which known ingredients react in a novel manner to produce a new substance or unexpected result.

of a monopoly over its use, manufacture or sale for a period of 17 years.[12]

Patent protection is, however, limited to the physical embodiment of the idea utilized by the invention -- not to the idea per se. The limited categories of patentable subject matter restrict patentability to functional interrelationships between physical things. Any process that depends for its novelty and unobviousness solely on a series of mental or intellectual steps is not patentable subject matter because it would create a monopoly over a law of nature or a disembodied idea.[13] Hence, abstract theorems, scientific principles, accounting or marketing methods and industrial know-how are all unpatentable because none of their inventive concepts is embodied in a physical form and no physical effects proceed directly from the operation of the theory, scheme or plan itself. A new chemical process, on the other hand, is an example of a process which, although not directly tangible or static, is patentable because it operates directly on and transforms physical constituents to different physical states or things. Since such a process produces a physical effect and accomplishes some change in the character or condition of the material objects on which it operates, it is a proper subject matter for process (as opposed to device) patentability.

Patent law requires disclosure of the underlying technology of the invention prior to the grant of monopoly protection. Before a patent will be issued, the inventor must file specifications by means of writings and drawings illustrating and describing at least one physical

---

12. The term of protection is the same in the Canadian and U.S. statutes. The protection is also renewable in each country for an additional 17-year term.

13. Section 28(3) of the Canadian Patent Act states that: "No patent shall issue for an invention that has an illicit object in view or for any mere scientific principle or abstract theorem." In Gottschalk v. Benson, the U.S. Supreme Court restated the "well established rule" in American patent law that: "A principle, in the abstract, is a fundamental truth; an original cause; a motive; these cannot be patented, as no one can claim in either of them an exclusive right... Phenomena of nature, though just discovered, mental processes, and abstract intellectual concepts are not patentable, as they are the basic tools of scientific and technological work." In In re Prater (415 F. 2d 1393 [C.C.P.A. 1969]), it was argued that if an integral part of any process or machine consisted of a series of mental steps that could be carried out wholly in one's head or by means of pencil and paper alone, the invention which employed it could not be patented. This became known as the "mental steps doctrine."

embodiment of the invention.[14] Disclosure is intended to serve two purposes: (1) to provide the public with access to the technology employed in the invention so that it can be used after the patent expires; and (2) so that others may ascertain whether their inventions can be employed or sold without fear of infringement.

There are fundamental differences between the types of protection afforded by patents and by copyright. First, a patent relates to a functional or physical embodiment which uses an idea, whereas copyright relates to an expression of the idea in the form of a literary, artistic or musical work. Copyright does not protect against independent discovery and development of the same idea in a different expression, whereas the physical embodiment of an idea in a patented invention can be protected even if someone else develops the same invention independently or alters its form. The requirement of originality for the purpose of copyright is much less stringent than the conditions of novelty, unobviousness and utility that a patentable invention must satisfy as a prerequisite for protection. Unlike copyright, patent protection is costly and time consuming to obtain because searches of previously filed claims must be conducted, and detailed specifications of the invention must be submitted.[15] Finally, the term of protection is significantly different in the case of patents and copyright. Copyright protection subsists, in most cases, for the life of the author plus 50 years, whereas a patent expires after 17 years.

Patentability of programs in software form

(a) U.S. position. The current position on program patentability in the United States has been described by one commentator as follows:

> If the state of program copyrights is confused, the state of program patents is utterly chaotic. The plot has all of the elements of a comic opera with four principal characters: the Patent Office, which steadfastly turns down every application for a patent on a computer program; the Court of Customs and Patent Appeals, which has fought for program patents in the face of increasing opposition from the Supreme Court; the Supreme Court, itself confused and trying to apply "nineteenth century

---

14. Patent Act, R.S.C., c. 203, s. 36 sets out the filing and specification requirements. The U.S. statute imposes similar requirements (35 U.S.C. s. 131 [1976]).

15. The average length of time between initial application for and the issuance of a patent now exceeds two years in both Canada and the United States. Total costs can exceed $1,000 for even the simplest of inventions. Anything more complicated would cost several times that much to patent. See Barrigar, 1976.

> legal notions to computer technology without understanding the technology" which keeps reversing the Court of Customs and Patent Appeals without directly confronting the issue of program patentability; and Congress, which, despite anxious pleas from the Supreme Court to solve the issue by statute does nothing. (Gemignani, 1980, p. 292)

Hence, despite three Supreme Court and numerous lower court rulings on program patentability the issue remains, for the most part, unresolved. Nevertheless, it is useful to examine the development of some of the case law in this area to derive general guidelines on what types of programs may or may not be patentable. As a prelude to the case history of program patentability, the following points should be kept in mind. First, as mentioned earlier, although abstract ideas or principles are not patentable, the machines or devices which employ those ideas or processes are. Second, one must distinguish between machine patent claims and process patent claims. Even if the idea or principle utilized by a machine is not novel or unobvious, a machine which utilizes that idea or principle in a novel or unobvious manner may be patentable. A process patent claim, on the other hand, requires novelty and unobviousness only in the process steps and not in the machinery that is to be used to carry out those steps.

Prior to the Supreme Court's landmark decision in Gottschalk v. Benson, the first of three cases on program patentability to be decided by that court, the U.S. history of program patents can be summarized briefly as follows. In 1964, the Patent Office issued a guideline stating that, in its opinion, programs were not patentable because they were mere abstract creations of the intellect (Puckett, 1968, p. 119). In 1965, the President's Commission on the Patent System recommended in its final report that:

> A series of instructions which control or condition the operation of a data processing machine, generally referred to as a program, shall not be considered patentable regardless of whether the program is claimed as: (a) an article; (b) a process described in terms of the operations performed by a machine pursuant to the program, or (c) one or more machine configurations established by a program. (1966, p. 12)

In other words, the Commission was rejecting program patentability in part because it rejected the definition of software which would make it part of a machine. It was also rejecting such a definition by rejecting program patentability. This recommendation was, however, never legislatively implemented.

In 1969 the Court of Customs and Patent Appeals (C.C.P.A.) rejected the Patent Office's guideline and the Commission recommendation

in the case of In re Prater (415 F. 2d 1393 [C.C.P.A. 1969]). That case involved a patent claim for a special apparatus and a computer program designed to analyze data from a mass spectrograph. The Patent Office rejected the claim on the grounds of what later came to be known as the mental steps doctrine: the process of analyzing the data could be carried out by hand using pencil and paper alone and hence the process was purely mental in nature and unpatentable. The Court of Customs and Patent Appeals overturned the Patent Office decision and granted "Prater's process" a patent. The Court stated that only those processes that were purely mental in nature in the sense that they could not be performed by a machine were unpatentable. Processes that utilized machinery to carry out the steps could qualify as patentable subject matter even if the novelty existed solely in the process rather than in the machine itself (415 F. 2d 1393 [C.C.P.A. 1969] pp. 889-890).

Shortly after its decision in the Prater case, the Court of Customs and Patent Appeals declared in In re Bernhart that programming a computer in a novel and unobvious way renders that computer physically different from the same computer without that program -- not in the sense that the program produces a new machine but rather because the program creates a "new and useful improvement" in an existing machine and hence falls within a permissible category of patentable subject matter (417 F. 2d 1395 [C.C.P.A. 1969] p. 1400). The basis of the C.C.P.A. decision was a slight variation from the definition of software as part of a machine. Had C.C.P.A. accepted the definition of software given here, the patent would not have been granted. The Bernhart case involved a system of connecting a computer to a plotter and using a set of equations with the system to plot various views of a three-dimensional object. These views could then be selected from any plane or from any point in space for visual display. The actual program was never specified in the patent claim; only the equations necessary for the transformations were submitted along with a description of their application. Furthermore, the only novelty in the claim was in the equations themselves since the program could be run on most general purpose computers. Nevertheless, Bernhart succeeded in obtaining a patent for his system.

In 1972, the United States Supreme Court decided Gottschalk v. Benson (409 U.S. 63 [1972]), its first computer program patent case. The case involved a computer program for converting binary coded decimal numbers to pure binary numbers. The applicant admitted that his method could be carried out by hand, although it would take considerably longer to do so. The patent claim was very broadly drafted; it purported to cover not only application of the method on computers but all possible applications. The Supreme Court held that Benson's process claim was unpatentable. Since the process consisted solely of a series of steps designed to solve a mathematical problem (the algorithm), it did not qualify as a patentable process within the meaning of the Patent Act. Moreover, Benson's claim was "so abstract and sweep-

ing as to cover both known and unknown uses of the binary coded decimal system to pure binary conversion" (Ibid., p. 67).[16] Since the system was not tied to any particular device or end use, granting a patent would be tantamount to granting Benson a monopoly over a general method of solving a mathematical problem. This the Court was not prepared to do, as patent protection would "wholly pre-empt the mathematical formula, and in practical effect would be a patent on the algorithm itself" (Ibid., p. 68). The Court's decision laid the foundation for rejecting the patentability of an algorithm by invoking the mental steps doctrine. The Court also restated the general conditions governing the patentability of processes utilizing mental steps which had been laid down in an earlier case, In re Abrams:

> 1. If all of the steps of a process claim are purely mental in character, the subject matter thereof is not patentable.
>
> 2. If a process claim embodies both positive and physical steps as well as so-called mental steps, yet the alleged novelty or advance over the art resides solely in one or more of the so-called mental steps, then the claim is considered unpatentable for the same reason that it would be if all the steps were purely mental in character.
>
> 3. If a process claim embodies both positive and physical steps as well as so-called mental steps, yet the novelty or advance over the art resides in one or more of the positive and physical steps and the so-called mental steps are incidental parts of the process which are essential to define, qualify or limit its scope then the claim is patentable and not subject to the objection contained in 1 or 2 above.[17]

Benson's claim was held to fall within the first category of process claims enumerated above. Hence, it failed. The Court specifically stated that it was not ruling out patent protection for computer programs in all cases but it failed to indicate under what circumstances the

---

16. Basically, Benson's program involved converting one representation of a number to another representation of that same number to facilitate the use of data on a computer.

17. Benson's algorithm could be performed on existing machinery and hence was not presented in terms of a machine patent claim.

tests of patentability would be met.[18] Nevertheless, the Court implied that if software is defined as a set of instructions, it would not allow patent protection of software, for a set of instructions to a computer is really nothing more than a representation of a set of mental steps.

Following the Benson decision, the Court of Customs and Patent Appeals (C.C.P.A.) placed an extremely limited interpretation on the holding in that case, reflecting its desire to continue granting patent protection for program-related inventions. In subsequent cases, the C.C.P.A. upheld patent claims where the program, when utilized with a particular device, was viewed as creating a new machine. For example, in the case of In re Noll (6 CLSR 69), the C.C.P.A. affirmed a patent claim for an apparatus for the display of text and other graphical information on a television screen for use in a computer graphics system. The system was described generally as a "programmable data processor operating under the control of a program" (Ibid., p. 69). The applicant conceded that the only inventive step resided in the program itself. Nevertheless, the C.C.P.A. upheld the patentability of the claim because, still considering a program as something other than a set of instructions, it viewed the applicant's machine, when programmed, as "structurally different from a machine without that program" (Ibid., p. 73). The court held that the Benson decision ruled out programs as patentable subject matter only if the claims were limited neither to any particular technology nor to any particular apparatus, machinery or specific end use. Since the claim in In re Noll was limited to a particular art or technology, tied to a specific apparatus and limited to a particular end use, the decision in Benson was held not to preclude patentability. It should be noted that, unlike Benson, In re Noll dealt with a machine patent claim as opposed to a process patent claim. The C.C.P.A. pointed out that such a distinction was crucial because protection was sought not for the program per se, as part of a process to perform a given task, but rather for a specific machine programmed in a particular and unobvious manner.[19]

---

18. The U.S. Supreme Court stated: "We do not hold that no process patent could ever qualify if it did not meet the requirements of our prior precedents. It is said that the decision precludes a patent for any program servicing a computer. We do not so hold." (409 U.S. 63 at 71).

19. This "new machine from a new program" argument is open to attack. Those who hold the view that a program does not turn a computer into a new machine point out that while a computer may be functionally "different" while operating under the control of a program because a specific sequence of operations is being carried out, the program does not physically modify the computer hardware. Recall also the earlier discussion of other flexible-use machines.

The second Supreme Court decision in this area was Dann v. Johnston, decided in 1976 (425 U.S. 219). The claims in Johnston were directed to an apparatus in the form of a specially programmed computer that enabled banks to perform financial record-keeping calculations for their customers.[20] The Patent Office rejected the claim on the grounds that, if a patent were granted, the applicants would have, in effect, a monopoly on that method of financial record keeping. The C.C.P.A. reversed the Patent Office decision, holding that Johnston's claim was tied to a particular apparatus and did not preempt the actual system of financial record keeping. Any bank would be free to use that system so long as they did not use Johnston's machine to do it. The Supreme Court reversed the C.C.P.A. ruling solely on the grounds that the data processor used to perform the calculations was "obvious." It did not deal with the patentable subject matter issue. The Court did not accept the contention that a general purpose computer is necessarily transformed into a "new" machine merely by being programmed in a particular way. The Supreme Court indicated once again that it was not deciding that programs were unpatentable under any circumstances (Ibid., p. 224). However, the Court again failed to spell out the circumstances under which programs might qualify as patentable subject matter. Subsequent to the decision in Johnston, the C.C.P.A. continued to reverse the Patent Office and found the following claims patentable: a computer-based control system for typesetting for use in conjunction with a conventional typesetter (In re Freeman, 573 F. 2d 1237 [C.C.P.A. 1978]); a computer device programmed to translate books from one language to another (In re Toma, 575 F. 2d 872 [C.C.P.A. 1978]) and a process utilizing a computer to regulate the operation of a system of multiplant units (In re Deutsch, 535 F. 2d 689 [C.C.P.A. 1977]).

The issue of software patentability was considered by the Supreme Court for the third time in the case of Parker v. Flook (437 U.S. 584 [1978]), which involved a patent claim for a method of updating alarm limits. The method consisted of a mathematical algorithm for automatically computing an updated alarm limit.[21] Computerized calculations producing automatic adjustments in alarm settings could then be generated by a means of program incorporating the algorithm. The patent claim was not tied to any specific computing device and covered a number of possible applications of the method. The Supreme Court held that the process was not patentable. Since the computation

---

20. Basically, the program was designed to sort cheques into prearranged categories to facilitate the recording of transactions.

21. An alarm limit is a number. During catalytic conversion processes, certain operating conditions, such as temperature, pressure and flow rate, must be monitored constantly. When any of these variables exceeds a predetermined alarm limit, an alarm will sound indicating a dangerous condition.

of updated alarm limits could be carried out using paper and pencil alone, the only novelty in the process resided in the algorithm and, as the formulation of the algorithm was purely mental in nature, the method could not qualify as a patentable process. The Court concluded that "a claim for an improved method of calculation, even when tied to a specific end use is unpatentable" (Ibid., p. 595). Once again, the Court was blocking the path of those wishing to define software as an algorithm; doing so meant that the software could not be granted patent protection. Instead the Court seemed to be defining software as a set of instructions and, as such, unpatentable. Furthermore, since the Flook claim purported to cover a broad range of the applications of the method, a grant of patent would have preempted the use of a mathematical relationship or "law of nature." The Court stated that the test for patentability was whether the process itself was novel and unobvious. Merely applying a new mathematical algorithm to carry out a known process was not sufficient to make the process patentable (Ibid., p. 591).

Once again, the Court declined to declare explicitly that all software is unpatentable, perhaps out of a concern that computer technology might change sufficiently in the future that some software might be patentable. In effect, however, the Court has denied patent protection to computer software because it defines software in such a way that the mental steps doctrine is invariably applicable. It is not clear that the Court fully realizes this effect of its decisions.

The current position of the C.C.P.A. on program patentability as set out in a number of post-Flook cases[22] decided by it can be summarized as follows:

1. Claims for a process involving programs which are essentially directed to a mathematical calculation and

---

22. Among them are In re Bradley (600 F. 2d 807 [C.C.P.A. 1979]); In re Sarkar (588 F. 2d 1330 [C.C.P.A. 1978]); In re Johnston (589 F. 2d 1070 [C.C.P.A. 1978]); In re Diehr (602 F. 2d 982 [C.C.P.A. 1979]); and In re Phillips (608 F. 2d 879 [C.C.P.A. 1979]). Most recently, "the U.S. Supreme Court has ruled [Diamond v. Diehr, No. 79-1112, 3/3/81] by a 5-4 vote that a process for treating synthetic rubber using a computer to calculate temperatures is an invention subject to federal patent protection. The Court said the proposed process used by Federal-Mogul Corp. of Southfield, Mich., is not simply a mathematical formula that would be excluded from patent protection." (The Globe and Mail, Wednesday, March 4, 1981.) Although this case sounds surprisingly like a reversal of Parker v. Flook, it could also fall under the fourth category listed in this study. Contrary to the statements in The Globe and Mail, it is not at all clear where the U.S. courts stand on the patentability of software. See also Diamond v. Bradley, No. 79-855, 3/9/81.

are not limited to any particular art or technology nor to any particular apparatus or machinery and which effectively preempt the algorithm are not patentable.

2. Claims for a process involving programs, even if tied to a particular machine, technology and end use, which provide for nothing more than a new and better system of calculation are not patentable.

3 Claims to a computer programmed in a novel manner, where the inventive advance lies in the machine itself rather than merely in the program, are patentable.

4. Claims to a method utilizing a computer program as part of an industrial process which involves some new physical application or post-algorithm solution, of which the actual solution of the algorithm is merely an intermediate part of the total process, are patentable.

(b) Canadian position. In contrast to the numerous decisions on program patentability in the United States, there have been but two reported decisions on the issue in Canada to date -- both decided by the Patent Appeal Board. Patent office policy in Canada is similar to that in the United States in that program-related inventions are looked on with general disfavour. Furthermore, the Economic Council of Canada in a 1971 report concluded that "patent protection of computer programs would not be appropriate" (p. 103). Nevertheless, in the first program patent case to come before the Patent Appeal Board, the Waldbaum case, decided in 1971, the Board upheld such a claim ([1972] 5 C.P.R. [2d] 162).

The invention in Waldbaum was presented in three different contexts: a method for controlling a data processor to determine the relative number of 0s and ls in a data set; a method of operating a data processor with specific application to counting the number of busy and idle lines in a telephone system; and a process comprising a new use of a computer. Hence, the claims in Waldbaum were drafted in terms of an apparatus, a new use of an apparatus and a process. The applicant conceded that his method was intended to be employed on a known data processor but argued that he had obtained an unexpected result since "it was not previously appreciated that this known computer could be made to operate in this manner until the applicant had devised this particular program" (Ibid., p. 166). Therefore "the programmed computer is a machine that is caused to operate in a new and unobvious way and is thus, a new and patentable machine" (Ibid., p. 166).

The Board, not having the benefit of the subsequent U.S. Supreme Court decisions in Benson, Johnston and Flook or any Canadian precedent to guide it, adopted the reasoning of the U.S. Court of Customs and Patent Appeals in In re Bernhart, a case which was subsequently overruled by the U.S. Supreme Court in Benson. The Board was satisfied that "a computer that is programmed in one way must be deemed to be a machine which is different from the same computer when programmed in another way or unprogrammed" (Ibid., p. 169). Waldbaum got his patent and the case was not appealed.

The next and only other case involving program patentability to come before the Board was Slomberger, decided in 1978.[23] The patent claim in Slomberger involved a computer program as part of a process to be used to calculate oil deposits. The applicant argued that his claim should succeed on the basis of the criteria set out in Waldbaum. The Board, in rejecting the claim, concluded that, in light of the developing American case law, its earlier decision in Waldbaum was no longer tenable:

> We are not satisfied, for the reasons given earlier, with all aspects of the Waldbaum decision because of more recent jurisprudence. For example, we are not satisfied that programming a computer in a particular way produces a new computer or indeed changes the computer in any way. It merely creates a temporary condition. A computer is inherently capable of performing a number of operations and in a particular sequence. No program can make a computer do something which it is not inherently capable of doing, because it is evident that general purpose digital computers are designed so that they are capable of responding to any program that can be devised to operate within the physical restraints of the machine. This is in fact the rationale in designing general purpose digital computers. Generally speaking programs are a kind of product that any competent programmer could produce, as a matter of course, using his normal skills. When a new program is produced nothing but intellectual information has been added to what previously existed. In our view any claim directed to it is not patentable, irrespective of whether the claim is directed to written instruction on how to operate a machine, or to an information carrier. (Patent Office Record, August 1, 1978, p. 25)

The Board agreed, however, with the dicta of the U.S. Supreme Court and with the current position of the U.S. Court of Cus-

---

23. The full text of the decision has not yet been reported fully. Portions of the decision appear in the August 1, 1978 issue of the Patent Office Record. Further details of the case were obtained from a telephone conversation with the chairman of the Patent Appeal Board.

toms and Patent Appeals in holding that, in certain circumstances, processes or apparatus utilizing programs might be patentable:

> It is clear however, that where an invention has been made in "a process control system" where a program is merely an incidental part of the system, it will not be objectionable. In a process control system you must however, have novel apparatus tied to a computer which controls a function at the end of a computer. In such a case the invention is not predicated solely on the novelty of the program. (Ibid., pp. 25-26)

The Board also suggested that the following criteria be adopted by the Commissioner of Patents when reviewing future patent claims involving computer programs:

1. Claims to a computer program per se are not patentable;

2. Claims to a new method of programming a computer are not patentable;

3. Claims to a computer programmed in a novel manner, expressed in any and all modes, where the novelty lies solely in the program or algorithm, are not directed to patentable subject matter under Section 2 of the Patent Act;

4. Claims to a computing apparatus programmed in a novel manner, where the patentable advance is in the apparatus itself are patentable; and

5. Claims to a method or process carried out with a specific novel computing apparatus devised to implement a newly discovered idea are patentable. (Ibid., p. 26)

In light of the Slomberger decision, it now appears that program patentability is governed by the same considerations in Canada as it is in the United States. Slomberger has, however, been appealed and is now pending before the Federal Court.[24] Until the Federal Court, and possibly the Supreme Court of Canada, have ruled on it, or until Parliament acts to make the legislation more precise, the law in this area will remain speculative.

---

24. A number of program patent cases currently pending before the Patent Appeal Board have been stayed, upon consent of the parties involved, until there has been a decision on the Slomberger appeal.

Summary. The present state of the law regarding the patentability of computer software is fairly clear in the United States. The Supreme Court has consistently invoked the mental steps doctrine to prevent patent preemption of algorithms. The Court has apparently decided that computer software is a set of instructions and is consequently unpatentable.

In Canada, the Patent Appeal Board has followed the U.S. experience quite closely. Initially, it accepted the definition of software as an integral part of a machine but, in light of the U.S. Supreme Court dicta, it has now reversed itself, holding that software per se is unpatentable. Despite the Patent Appeal Board's reasonably clear statement on the unpatentability of software, whether computer software can be patented here is still an unsettled issue because the courts have yet to rule on the question.

With the invocation of the mental steps doctrine to deny the patentability of computer software, the United States Supreme Court and the Patent Appeal Board in Canada have effected a policy which can be discussed in terms of the rationales set out earlier in this chapter. These bodies have in other decisions recognized that, while granting a temporary monopoly for the output of creative effort may (and only may) be detrimental in the short run, it generally benefits society in the long run by encouraging creative effort. Nevertheless, granting such temporary monopolies for scientific principles or for abstract theorems will, it is feared, have detrimental social effects by foreclosing them or at least making their use more costly. It has apparently been decided that the social costs of granting even a temporary monopoly for ideas outweigh the social benefits of encouraging the production of ideas by granting the monopoly. The North American economies have instead tended to encourage the production of ideas via the marketplace or direct government grants. The pros and cons of patent protection for computer software will be discussed in greater detail in Chapter III; the remainder of this chapter describes other possible legal protection for software.

Copyrightability of Computer Programs

Jurisprudential background of copyright. The primary justification for copyright is the conviction that encouraging individual literary and artistic effort by personal gain is the best way to promote public welfare. As pointed out at the start of this chapter, copyright, by conferring a limited monopoly, is intended to encourage authors to write and publish by removing the fear of plagiarism. Hence, the individual author or artist will be ensured of being adequately compensated for his or her efforts while society will benefit through publication of the work.

Since the Canadian Copyright Act came into force on January 1, 1924, it has not been revised; only minor amendments have been made

from time to time. The Copyright Act is a federal statute since the B.N.A. Act confers exclusive jurisdiction over "copyrights" to the federal government (s. 91 [23]). The basic legal principles implicit in our copyright statute and developed by the case law can be stated briefly as follows. Copyright in Canada is automatic in all published or unpublished works (s. 4). Unlike the U.S. legislation, which requires the affixing of a standard copyright notice to each published copy of the work, Canadian copyright protection subsists independently of registration or any other formal act. The only requirements are that the author be, at the date of making of the work, a citizen or subject of Canada or of a country with which Canada has an international agreement.[25] Any "original artistic, dramatic, musical or literary work" qualifies for protection (s. 2). Copyright is the negative right of preventing the copying, in any medium, of original works existing generally in the fields of literature and the fine arts. The protection afforded by copyright is inherently narrow -- the prohibition is only against copying form. There is no copyright in ideas, information or plot.[26] Furthermore, copyright protects only against copying and not against similarity or duplication arising from independent creation or access to common research sources.[27] Copyright extends only to the form of the work and not to the physical material embodying the work.[28] The requirement of an original work does not imply that the work must possess originality in a qualitative sense; "original" simply means that the work has not been copied.[29] Because ideas, information and plots are not generally copyrightable, copyright subsists only in a work that has been reduced to a tangible, fixed form.[30] In general, the term of copyright protection is the life of the author plus 50 years

---

25. For example, through the Berne or Universal Copyright Conventions to which Canada is a signatory.

26. See Goldner v. C.B.C. ([1973] C.P.R. [2d] 158) and Collins v. Rosenthal, ([1974] 14 C.P.R. [2d] 143). Despite the general lack of protection for ideas, the prohibitions against adaptations do, however, appear to protect plot.

27. Collins v. Rosenthal ([1974], 14 C.P.R. [2d] 143).

28. Webb & Knapp (Canada) Ltd. et al. v. City of Edmonton ([1970] 63 C.P.R. 21).

29. Kilvington Bros. Ltd. v. Goldberg et al. ([1957] 16 Fox P.C. 164).

30. Stevenson v. Crook ([1938] Ex C.R. 299). Barry Torno has emphasized to the authors that the lack of protection for ideas is not at all clear.

(s. 5).[31] Generally, the first owner of copyright is the person who writes, draws or composes the work.[32] In the case of an employee who produces the work pursuant to his contract of employment, the employer is the owner of the copyright (s. 12[3]). Registration of a copyright merely serves to provide prima facie evidence of the owner of the copyright; registration per se creates no substantive rights (s. 36[1]). Copyright is infringed when one copies the work or a substantial part thereof in any medium without the permission of the owner. This rule is, however, subject to many statutory exceptions and to the doctrine of "fair dealing."[33] Finally, the Copyright Act permits the issuance of compulsory licences to publish copyrighted works in certain circumstances (sections 14,19).

With this brief description of the function of copyright and its underlying legal principles, the particular problems that computer programs have presented can be examined in an attempt to fit them within the scope of copyrightable subject matter.[34] Since there is, as yet, no reported decision in Canada dealing with the applicability of copyright to computer programs, the discussion of this issue will be confined to the legal position as it has evolved in the United States.

The legal position in the United States is relevant for Canada since the legal principles and the philosophical basis of the U.S. copyright legislation are very similar to those in Canada.[35] Moreover, when and if the issue of programs and copyright is litigated in Canada, the Canadian courts will undoubtedly look to the American jurisprudence for guidance.

Particular problems in applying copyright to computer programs. A major revision of U.S. copyright law was effected by the federal Copyright Act of 1976 which came into force on January 1, 1978. This new statute was the end result of several years of prolonged study and intense debate. The basic doctrinal principles of copyright law were, for

---

31. The term for sound recordings of musical works is fifty years.

32. Evans v. Hulton & Co. Ltd. ([1924] L.T. 534).

33. Examples of acts not constituting infringement are enumerated in s. 17(2) of the Act.

34. There is a substantial volume of literature on this area. Some of the better writing includes: Puckett, 1968; Stork, 1968; Gemignani, 1980; Breyer, 1970; Barrigar, 1976; Nelson, 1966.

35. In discussing the legal position in the United States, any differences in the Canadian legislation will be pointed out in the footnotes where appropriate.

the most part, left untouched by the new Act; its purpose was to make some definitional changes, to clarify interpretations of certain principles and to make appropriate modifications to the old legislation to adapt it to the technological changes that had occurred in the production and distribution of copyrightable subject matter through such means as computer data information systems, reprography, video communications, sound recordings and library network and information systems.[36]

Although the old legislation did not expressly mention computer programs as a copyrightable work, the Register of Copyrights has, since 1964, accepted programs for registration under the classification of "books and other literary works" upon the following terms:

> The registrability of computer programs involves two basic questions: (1) Whether a program as such is the "writing of an author" and thus copyrightable, and (2) whether a reproduction of the program in a form actually used to operate or be "read" by a machine is a "copy" that can be accepted for copyright registration. Both of these are doubtful questions. However, in accordance with its policy of resolving doubtful issues in favour of registration wherever possible, the Copyright Office will consider registration for a computer program if certain requirements have been met....
>
> Registration for a computer program will be considered if:
>
> (a) The elements of assembling, selecting, arranging, editing and literary expression that went into the compilation of the program are sufficient to constitute original authorship.
>
> (b) The program has been published, with the required copyright notice....
>
> (c) The copies deposited for registration consist of or include reproductions in a language intelligible to human beings. If the first publication was in a form (such as machine-readable tape) that cannot be perceived visually or read by humans, something more (such as a print out of the entire program) must be deposited. (Computer Programs, Cir. 31 D, 1965; CONTU, 1978, p. 85)[37]

---

36. For a survey of the developments leading to the enactment of the new copyright statute, see Stedman, 1976.

37. See also Tapper, 1978.

Despite the willingness of the Copyright Office to accept programs for registration, copyright has not been a much sought after form of protection by the software industry. As of January 1, 1977 only 1,205 programs had been registered out of the approximately 1,000,000 to 4,000,000 programs estimated to have been developed annually.[38] Furthermore, there are but two reported decisions in the case law directly touching on the validity of a program copyright.[39]

Although the Copyright Act of 1976 does not specifically mention computer programs in its provisions, there is substantial evidence that Congress intended that computer programs lie within its ambit. For example, the official commentary to the Act contained in the House and Senate reports accompanying the introduction of the legislation expressly states that programs are copyrightable "to the extent that they incorporate authorship in the programmer's expression of original ideas, as distinguished from the ideas themselves" (H. Rep. No. 1476, 94th Congress, 2nd Session, 1976, p. 54). It should be noted, however, that the new Act will not, nor was it intended to, change to any great extent the availability of copyright protection to computer programs as it may have existed prior to January 1, 1978. Section 117 of the new Act provides that:

> Notwithstanding the provisions of sections 106 through 116 and 118, this title does not afford to the owner of copyright in a work any greater or lesser rights with respect to the use of the work in conjunction with automatic systems capable of storing, processing, retrieving, or transferring information, or in conjunction with any similar device, machine, or process, than those afforded to works under the law whether title 17 or the common law or statutes of a State, in effect on December 31, 1977.

In addition, a House Report makes the following comments on section 102 of the Act, which defines copyrightable subject matter for the purpose of that statute:

> The purpose of this added language [in s. 102] was not to enlarge or contract the scope of copyright protection under the present law, but rather "to restate, in the context of the new single Federal system of copyright, that the basic dichotomy between expression and idea remains unchanged." The new provision had been added

---

38. CONTU, 1978, p. 85 and Tapper, 1978.

39. Synercom Technology, Inc. v. University Computing Co. (462 F. Supp. 1003) and Data Cash Sys. Inc. v. JS & A Group, Inc. (480 F. Supp. 1063). These cases will be analyzed later in the study.

> in response to the great debate over computers and copyright, and is intended to disclaim any intention to protect a programmer's algorithms under the bill. (H. Rep. No. 1476, 94th Congress, 2nd Session, 1976, p. 116)

Hence, except for some strong indications of Congressional intent that computer programs should, in some cases, qualify for copyright protection, the Copyright Act continues to leave unresolved the issue of the applicability and extent of protection for computer programs. It does, however, contain expanded definitions of what constitutes a "writing" and a "copy" for the purpose of copyright law.[40] These expanded definitions might make it easier for programs to fall within the ambit of copyrightable subject matter.

Next, the problems of fitting computer programs within the scope of copyright protection are examined.

(a) Can computer programs qualify as "literary works fixed in a tangible medium of expression"? The National Commission on New Technological Uses of Copyrighted Works (CONTU), a body set up by the U.S. government to examine, among other things, questions concerning computer uses of copyrighted works, argued in its final report that programs should be treated for copyright purposes as "literary works" (1978, p. 85). Section 102 of the Copyright Act of 1976 extends protection only to "original works of authorship fixed in any tangible medium of expression, now known or later developed, from which they can be perceived...either directly or with the aid of a machine or device."[41] Literary works are defined in section 101 to include works "expressed in words, numbers, or other verbal or numerical symbols or indicia, regardless of the nature of the material objects, such as books, periodicals, manuscripts, phonorecords, film, tapes, disks, or cards, in which they are embodied." (Emphasis added to show that this list is not intended to be comprehensive.) An "author" for the purpose of the copyright legislation is the person who created the work. The requirement of originality is satisfied by the mere lack of copying; the term "original" does not require any degree of ingenuity or unobviousness (Puckett, 1968, p. 119). If one accepts the definitional approach of a program as a descriptive set of instructions expressed in language or symbols, then it appears likely that a program will qualify as a literary work for the purpose of copyright. The term "literary works," in its technical sense, does not connote any criteria of literary or qualitative merit. Catalogues, telephone books, maps and charts and compilations of data have all traditionally been proper subjects of copyright. Furthermore, programs committed to punched cards, paper, disks or

---

40. These definitions will be considered later.

41. This paper recommends in Chapter III that the phrase beginning "from which they can be perceived" be deleted.

tapes or in the computer memory which, when fed into the computer can be translated to language or symbols by means of a printout, probably do meet the requirements of fixation and perception that the statute prescribes. In addition, there is no doubt that the supporting software material that is frequently supplied as part of the program package to users, such as descriptive instruction manuals and other documentation which facilitate the use and adaptation of the actual program, are properly copyrightable.

There are, however, cogent arguments against classifying computer programs as literary works. Essential differences exist between programs and other works of authorship covered by copyright. One crucial distinction that has been pointed out by several commentators is that computer programs are not similar to ordinarily copyrightable sets of instructions for mechanical tasks.[42] The instructions in a computer program, unlike other printed instructional material, are not directed towards human understanding or perception but rather are created to enable a particular machine to function. One member of CONTU, Commissioner Hersey, attacked in his dissenting report the view of the majority report which recommended inclusion of programs within the scope of copyright protection by classifying them as literary works.

Quoting from that dissent:

> An argument commonly made in support of the copyrightability of computer programs is that they are just like ordinary printed (and obviously copyrightable) lists of instructions for mechanical work. The software subcommittee calls a program "a writing which sets forth instructions or sets of instructions." But this analogy or metaphor, does not hold up. Descriptions and printed instructions tell human beings how to use materials or machinery to produce desired results. In the case of computer programs, the instructions themselves become an essential part of the machinery that produces the results...when it comes to the object, or operative, stage of the program, when it becomes indistinguishable from a machine part and does work -- all resemblance to a "writing" or "description" has been lost. It is a machine tool. In use it has become part of a mechanical process. It is not copyrightable....The Software Subcommittee Report recommends affording copyright protection to something that starts as a writing but ends as a labour saving mechanical device. (CONTU, 1978, p. 77)

---

42. Most notably in the dissent of Commissioner Hersey in the CONTU final report, 1978. See also the discussion by Melville, 1980.

Commissioner Hersey's arguments are not convincing. As a result of changing technology, the computer will increasingly be used as an intermediary in the production of literary and cultural works. For example, programs exist which facilitate translation from one language to another and music is also composed using computers. More important, however, is the fact that whether computer software qualifies for consideration as literary work is irrelevant for policy prescription. If analysis suggests that computer software should be provided copyright protection, it should not matter whether it is classified as literary work or if a new category of copyrightable subject matter is created to include computer software. As an example, some people might argue that maps, compilations and directories are not literary works as the term is commonly understood; nevertheless the Copyright Act simply extends the definition of literary works to include these items because society has decided that they should be included in the scope of copyright protection. Thus, it doesn't matter what computer software is called; so long as the analysis here leads to a recommendation that it be included in the Copyright Act, the new legislation must specify that decision clearly.

Commissioner Hersey's comment may, though, be part of a broader criticism of expanding the scope of copyright protection to include computer software. Copyright has generally been reserved for cultural creative effort as opposed to more utilitarian, mechanical or technological creative effort. His dissent may reflect concern that, if the traditional coverage of copyright were expanded to include such items as computer software, the traditional basis for copyright -- the encouragement of the production of cultural output -- would be altered so drastically as to lose its importance. Opposed to this concern is the belief that, unless recognition is granted to new technologies, creative effort may be impeded without appropriate legislative intellectual property protection. At the beginning of their study, Braunstein et al. address these concerns directly:

> Since the passage of the [U.S.] 1909 Copyright Act (and in some ways, since the enactment of the first U.S. copyright law in 1790) it has been necessary for the courts and the Copyright Office to interpret that act in the light of technological and business advances. Two trends have emerged. First, utilitarian (non-artistic) creations and compilations of data such as interest tables and telephone books have been found to be copyrightable. Second, the relationship of the Copyright Act to the products of and information transmitted by new technologies has frequently created difficult cases and often bad, or at least unclear, law. Among these technological advances have been sound recordings, radio and television broadcasts, photocopying, and cable television. (Braunstein et al., 1977, p. I-1)

In other words, this paper's response to these concerns is two-fold. First, the present Copyright Act is not strictly concerned with what might commonly be interpreted as cultural activity. Second, even if expanding the Act to include computer software involved expanding its traditional scope, this would break with tradition only in the scope and not in the past interpretation of the Act. Such a break with tradition (if it is a break at all) would be quite different from allowing the copyrightability or patentability of algorithms (a different type of break with tradition which is rejected in Chapter III); the latter would involve changing the interpretation of intellectual property law as it applies in all areas whereas the former would only extend current interpretations of the law.

(b) The distinction between form of expression and underlying ideas -- the problem of proof of infringement. Assuming that copyright protection can be made to extend to a computer program, there remains the problem of determining the exact nature of the protection to be afforded. The problem is due to the inherent narrowness of copyright which generally protects only the form of expression but not the underlying ideas in the work. Moreover, copyright does not prohibit others from producing separate and independent works utilizing similar concepts. Section 102(b) of the U.S. Copyright Act codifies the longstanding principle of copyright law:

> (b) In no case does copyright protection for an original work of authorship extend to any idea, procedure, process, system, method of operation, concept, principle, or discovery, regardless of the form in which it is described, explained, illustrated or embodied in such work.

With respect to computer programs, the problem of distinguishing between a program's protected form of expression and its unprotected underlying idea(s), methodology or logic is particularly problematic. This difficulty exists because, unlike books or movies in which the form of expression has independent value in a cultural or aesthetic sense, the particular form in which a computer program is expressed has little value per se apart from its ability to cause a computer to function in a particular manner. Colin Tapper described this peculiar feature of programs as follows:

> In a sense identity of form is more important in the case of a computer program in that a minute variation such as the transcription of a comma and space might well result in total failure. On the other hand much more substantial re-writing and re-arrangement deriving from close scrutiny of the original might well result in complete success. (Tapper, 1978, p. 17)

Hence, if the program's primary utility lies in its main underlying idea (i.e., the methodology or algorithm) and not in what form the program

happens to take (i.e., its translation into a particular computer language and onto a particular medium), then copyright protection of the format will create little, if any, monopoly power for its author. The form of a program may often be disguised easily and cheaply while, at the same time, retaining the same logical sequence of steps or algorithm as the original.[43] Computer hardware technology is such that it is usually possible to program a computer using different formats and still arrive at the same result. Hence, it might legally be permissible for one person to appropriate the algorithm from a competitor's program and create his own, which would appear on its face to be significantly different. This might be done by leaving the algorithm from the original program intact but changing the sequence of data, variable names, order of instructions, statement numbers, computer language and the format of the output for the "new" program.[44] Both programs would have the same underlying logic and both could be used to perform the same function. The format of the two programs might, however, be radically different in appearance. It should be noted that the "copier" would have to do more than merely translate the original program into a new computer language, since a translation of a work into another language would most likely constitute copyright infringement.[45] It should also be noted that the process of debugging a new program, even if it incorporates a well-known algorithm, is by no means costless.

Copyright, then, would not serve to prevent the appropriation of programming ideas. Plagiarism of a program's form would be within the scope of copyright protection but the ease with which that form can be disguised and the immense difficulties in proving sufficient similarity to raise an inference of copying severely weaken this form of legal protection. The problem is rendered even more onerous by the fact that proof of similarity is not enough. The alleged copier will invariably raise the defence that he created the similar work separately and independently. Hence, the plaintiff in such an infringement action will have to prove that the defendant had access to the work. Furthermore, for programs utilizing simple algorithms the courts may deny copyright protection altogether, even if the program's form has been exactly duplicated, on the grounds that the number of ways in which such

---

43. Gemignani (1980) points out that teachers of computer science courses have difficulty detecting whether two students have helped one another in writing the same programming assignment.

44. Recall that any given program may be coded according to any number of different programming notations or languages.

45. In both the Canadian and the U.S. legislation, an author has the exclusive right to translate the copyrighted work into other languages or dialects. Assuming that the analogy between linguistic and computer languages stands up, the mere translation of a program to another computer language would consititute infringement.

simple concepts can be expressed is strictly limited. For example, in Continental Casualty Co. v. Beardsley, which dealt with an insurance company's use of a lawyer's copyrighted forms, the court stated that:

> the use of specific language may be so essential to accomplish a desired result and so integrated with the use of a...conception that the proper standard of infringement is one which will protect as far as possible the copyrighted language and yet allow the free use of thought beneath the language. The evidence here shows that [the company] insofar as it has used the language of [the lawyer's] forms has done so only incidental to its use of the underlying idea....In so doing it has not infringed. (253 F. 2d 702 [2nd Cir. 1958])[46]

Finally, the probability of detecting unauthorized use is particularly low in the case of computer programs. A copier of books or movies can generate financial gains only by reselling them. Computer programs, however, need not be resold by a copier to be of any value to him. The copier can derive substantial financial benefits from the program by appropriating it and storing it within his own computer memory for use on his machine. One problem with software copyrightability, then, would be the role of fair-dealing defences (discussed in the section on copyright protection in Chapter III).

The problem of the proof of infringement is similar to the problem of proof of violation of trade secrecy protection, discussed in the section on trade secrets and computer programs later in this chapter. In both cases, a question arises as to how important the benefits of the protection would be if misappropriation is difficult to detect or establish.

(c) What constitutes a copy of a computer program? Copyright does not generally prevent the mere unauthorized use of a pirated work; infringement arises from copying, not from simply using the work. For example, the baking of a cake is not an infringement of the copyright in the recipe nor is the playing of a game by following a set of copyrighted rules an infringement. In general, infringement occurs through the use of a copyrighted work only if a copy of the work is generated as a result of such use.[47] Assuming that a program is copyrightable, would the mere unauthorized inputting of the program in its original, unduplicated form into a computer constitute infringement? The answer depends on whether or not the use of the program has resulted in copying. The definition of "copies" is extremely broad under the new U.S. Act. Section 101 provides the following definition:

---

46. No similar case decided in Canada could be located.

47. Public performances of copyrighted works are a rare exception to this general proposition.

> "Copies" are material objects, other than phonorecords, in which a work is fixed by any method now known or later developed, and from which the work can be perceived, reproduced, or otherwise communicated, either directly or with the aid of a machine or device. The term "copies" includes the material object, other than a phonorecord in which the work is first fixed.[48]

The predecessor 1909 Act required a copy to be visually perceptible to, or capable of being read with, the naked eye. This earlier restricted definition led to decisions that piano rolls and records were not copies of the sheet music from which they were derived since they possessed no visible intelligibility.[49] The new Act has therefore considerably expanded the notion of what constitutes a copy to include any material object that can be perceived in any manner, whether visibly or otherwise, "either directly or with the aid of a machine or device" (s. 101).

Moreover, a work in one medium that has been copied from a work in another medium does not necessarily lose its character as a "copy" for the purpose of copyright. Motion pictures are held to be copies of plays (s. 3[1][e]), dolls have been held to be copies of cartoons[50] and photographs copies of sculptures.[51] This principle was, however, subject to one qualification. If the copy of the work was embodied in the form of a "useful article" no infringement took place. For example, in the case of Mazer v. Stein (347 U.S. 201) no infringement was found where the defendant manufactured lamps using the designs he found, in pictorial form, in the plaintiff's copyrighted catalogue. Nor is it an infringement to construct a bridge or a building based on a

---

48. It is an open question whether the definition of copy in the Canadian legislation is as broad as the one contained in the U.S. statute. This turns on the interpretation of section 3 in the Canadian Copyright Act. A copy has been defined judicially in English cases as "that which comes so near to the original as to give to every person seeing it the idea created by the original" (King Features Syndicate Inc. v. Lechter [12 C.P.R. 60] citing with approval the case of West v. Francis [(1822) 5 BE Ald. 737]).

49. White Smith Music Publishing Co. v. Appolo Co. (209 U.S. 1 [1908]).

50. King Features Syndicate, Inc. v. Fleisher (299 Fed 533 [2nd cir. 1924]).

51. Bracken v. Rosenthal (151 Fed 136 [C.C.N.D., Ill. 1907]).

set of copyrighted drawings and plans.[52] Section 113(a) of the U.S. Act now extends protection to pictorial, graphic and sculptural works when they are embodied in the form of "useful articles" and so changes the law in this respect insofar as the original work can be classified as "pictorial," "graphic" or "sculptural." However, so far as the use of copyrighted works in conjunction with computers is concerned, a close reading of section 117 suggests that the benefit of section 113(a) will not be available to programs.[53]

Hence, the current U.S. legal position with respect to both when copying of a computer program occurs and what constitutes a copy appears to be as follows. First, someone who duplicates the original source program onto another medium, such as paper, disk or tape, prior to the input or at the output stage has probably made a copy of the program within the expanded definition contained in section 101. A program on disk or tape, like the music on a record, though not directly perceptible can be made so with the aid of a "machine or device" -- for example, by means of a printout of the actual program. Second, the actual unauthorized inputting of the original program into a computer does not constitute copying per se except that a printout generated in the input process or at any other time would probably constitute a copy. Although use of the program may involve its reproduction in the computer memory, such internal reproduction is not sufficient to constitute a copy.

This peculiar feature of U.S. legislation has resulted because, unlike the case of duplication of a program prior to its use on a computer, the actual use of copyrightable material in conjunction with a computer continues to be governed, by virtue of the "freezing" provi-

---

52. Muller v. Triborough Bridge Authority (43 F. Supp. 298 [1942]). The Canadian law differs with respect to architectural plans. In Canada, the construction of a building according to an architect's plans without his/her authorization is a reproduction of the work in a material form and hence constitutes infringement (sections 2(a), 3). In addition, section 46 of the Canadian Copyright Act excludes from the statute those works eligible for industrial design protection under the Industrial Design Act. If an article is to be reproduced in more than 50 copies, it is presumed to be an article of manufacture and the Industrial Design Act rather than the Copyright Act applies. Hence, if a work that would ordinarily qualify for copyright protection becomes embodied in an article of manufacture, the only protection available against the copying of that work in the new medium is industrial design protection.

53. Section 117 states that the use of copyrightable material in conjunction with a computer continues to be governed by the old law.

sion in section 117, by the old law. This prior law, as mentioned earlier, required a copy to be visibly intelligible without the aid of a machine or device. Internal reproduction in the computer memory would clearly not meet this requirement. Similarly, simply reading a program from one computer to another by transmitting, electronically, the pattern of switch settings from one machine to another would probably not constitute copying. Finally, a program that has been hardwired in object form or in the form of a silicon chip which, when inserted into the computer, activates the programming instructions internally, probably does not qualify as a "copy" of the original source program from which it was derived, for two reasons. First, the silicon chip incorporating the program in its circuitry can be classified as a "useful article" and hence copying the program in this form may be permissible.[54] Secondly, when the chip is actually used in conjunction with the computer, no visibly intelligible copy of the program is normally generated. Hence, by virtue of section 117 and the case law prior to 1978, no "copy" would be generated by its use.

Case history. There is a paucity of case law on the issue of copyright infringement of computer software. To date, there are no Canadian cases and but two U.S. cases reported that have dealt with this legal question directly. Both are lower court decisions. Hence, the law in this area will remain a subject of conjecture until higher courts and/or legislative bodies have had an opportunity to deal with it. Nevertheless, both cases are interesting illustrations of the possible "thinness" of copyright protection and the difficulty that judges have in ascertaining the dividing line between form of expression and ideas and answering the question of what constitutes a copy for the purpose of copyright protection for computer programs.

(a) Synercom Technology, Inc. v. University Computing Co. (462 F. Supp. 1003 [N.D. Tex. 1978]). Synercom, the plaintiff, had developed a computer program designed to solve engineering problems incident to the analysis of structures. It had also developed a unique format for the input of data for use with the actual program. Synercom had spent considerable time and money to conceive the logic and arrangement of entering engineering data into a format that greatly facilitated and expedited the required calculations. The program it developed was specifically designed to accept the data in this format. In addition,

---

54. In Canada, if more than 50 silicon chips were manufactured it would appear that section 46 would exclude copyright protection since industrial design protection would be available. Industrial design protection would, however, be useless for silicon chips. The Industrial Design Act protects only the ornamental or visually distinct aspects of an article -- not its functional or structural nature. Since one chip looks very much like any other chip it is doubtful whether industrial design protection is of any benefit. Unless clarified in statute, a Catch-22 situation could easily evolve.

Synercom wrote a lengthy instruction manual which it supplied to customers explaining and illustrating the methodology of transforming the raw data into the required format, together with examples of specific applications of the program itself. The defendant, University Computing Company, had prepared a preprocessor program from the descriptions contained in Synercom's instruction manual. This program was designed in a manner that permitted it to process data by using the same input format developed by Synercom. Users of Synercom's software could therefore convert to the defendant's program without a corresponding need to change their already familiar input format for the entry of data. It should be noted that the defendant had no access to the actual program developed by Synercom but only to Synercom's instruction materials, which contained the description of the input format and programming applications.

Synercom brought suit for copyright infringement of its instruction manual and input format. Although Synercom did not allege copyright infringement of the program itself, the case is relevant on this issue because programs and input formats are very similar in their development, preparation and appearance, and because the judge in the case commented specifically on the copyrightability of programs.

The court held that Synercom's input formats were not copyrightable. Such formats could properly be the subject of copyright only if the ideas they expressed were separable from their expression. The court concluded that the order and sequence of software formats are not expressions of ideas but are rather the ideas themselves:

> The difficult question is whether [the defendant] plagiarized Synercom's idea or its expression. If the idea is the sequence and ordering of data there was no infringement. If sequencing and ordering of data was, however expression, it follows that the defendant's program infringed. As earlier suggested and as will be demonstrated, Synercom's argument is double-edged. If sequencing and ordering is expression, what separable idea is expressed? (Ibid., p. 1013).

Commenting specifically on the possibility that the defendant infringed the actual program the judge noted that:

> The preparation of a computer program in any language from a general description of the problem to be solved (as, for example, is contained in the forms and manuals, which prescribe a problem involving a set of ordered inputs in a particular arrangement which must be accepted to the FORTRAN program) is very dissimilar to the translation of a literary work, or to the translation of a program from one language to another. In most cases, the formulation of the problem in sufficient detail and

> with sufficient precision to enable it to be converted into an unambiguous set of computer instructions requires substantial imagination...and the resulting program can in no way be said to be merely a copy or a version of the problem statement. The program and the statement are so different, both in physical characteristics and intended purpose, that they are really two different expressions of the same idea, rather than two different versions of the same expression. Hence [the defendant's] preparation of a FORTRAN preprocessor program from the descriptions contained in the manuals cannot constitute an infringement derivative use provided this was done without copying of the plaintiff's [actual] program. (Ibid., p. 1013, n. 5)

Although the defendant had not infringed the input format because it was held uncopyrightable, in informing its customers on how to use the input format with its new program it had copied extensively from Synercom's copyrighted instruction manual. Hence, Synercom succeeded in proving infringement with respect to its copyrighted instructional material and was entitled to a remedy on this ground. Had the defendant rewritten the instruction manual in its own words, rather than slavishly copying Synercom's material, there would have been no finding of infringement.

(b) Data Cash Systems, Inc. v. JS & A Group, Inc. (480 F. Supp. 1063 [N.D. Illinois 1979]). The plaintiff, Data Cash Systems, Inc., had retained a firm of independent consultants to design and develop a computer program for a computerized chess game, Compu Chess, which it intended to manufacture and market. The program consisted of a set of instructions which enabled the computer to play chess at six different skill levels. These instructions were translated into programming language which was in turn translated into machine language by means of an object program. This object program, known as the Read Only Memory (R.O.M.), was hardwired into the computer as part of its internal circuitry. Thus, Compu Chess was a hand-held computer by which the user entered his moves on a keyboard device and the computer relayed its move by means of the internal object program. This move was visually displayed to the player in a manner similar to the visual display devices on hand-held electronic calculators. The source program for this game was filed with the Register of Copyrights and a certificate of registration was issued to Data Cash in November 1978. The source program was not made available to buyers of Compu Chess since the computer program was fully hardwired. Shortly after the plaintiff began marketing Compu Chess, the defendant, JS & A, began marketing the JS & A Chess Computer game. The R.O.M. in this game was identical to the R.O.M. in Compu Chess and this was admitted by the defendant when Data Cash sued for copyright infringement.

In its judgement, the court assumed and accepted that there had been direct copying of Data Cash's R.O.M. Nevertheless, the defendants were not held liable for infringement because Justice Flaum concluded that a copy of a program in R.O.M. form did not constitute a copy for the purpose of copyright law. The expanded definition of "copy" in the 1976 Act was held not to apply because section 117 of that Act stipulated that the extent of copyright protection afforded to subject matter used in conjunction with computers was to be determined by reference to the prior legislation. In examining the requirement for a copy for the purpose of the copyright law as it existed prior to 1977, Justice Flaum concluded:

> That the R.O.M. at common law does not constitute a copy of the plaintiff's computer program is supported by the cases which hold that a completed building is not a copy of the architectural plans upon which the building is based. An architectural plan is a technical writing which is capable of being copied only by other similar writings, i.e., by other plans. A building is the result of plans not a "copy" of them. It follows that a copy of a computer program is another computer program in its flow chart or source phase because these are comparable technical writings. While the ROM is the mechanical embodiment of the source program it is not a "copy" of it. (Ibid., p. 1068)[55]

Even if the expanded definition of copy in the 1976 Act was applicable, the court stated that copying of the R.O.M. would not be actionable because "...in its object phase, the computer program is a mechanical device which is engaged in the computer to become an essential part of the mechanical process. Mechanical devices which cannot qualify as pictorial, graphic or sculptural works are not writings and may not obtain copyright protection" (Ibid., p. 1067, n. 4).

Justice Flaum's comparison of copying a competitor's R.O.M. as analogous to constructing a building from a set of architectural plans is not entirely apt.[56] Constructing a building from a set of plans involves translation into another medium whereas, in this case, JS & A had copied its R.O.M. directly from the Compu Chess R.O.M. In other words, JS & A had not translated Data Cash's R.O.M. into or from a different medium. Secondly, it is submitted that Justice Flaum misinterpreted section 117 of the 1976 Act. As mentioned earlier, this section directs that the prior law of copyright be applied only in cases

---

55. Justice Flaum's analogy is inappropriate for Canadian law. (See footnote 52.)

56. The Canadian law is different with respect to architectural plans. (See footnote 52.)

where the act of copying occurs as a result of the use of the subject matter in question in conjunction with the computer. Where, as in the Data Cash case, copying occurs prior to and not as a result of the use on a computer, the expanded definition of copy, section 101 of the 1976 Act, governs. It would have been better to hold that a R.O.M. is not a copy of a program because it is mechanically embodied in the form of a useful article. Moreover, the R.O.M. per se is not copyrightable because it is not a literary, pictorial, graphical or sculptural work. The recommendation in Chapter III that there be explicit copyright protection for firmware is intended to avoid the risks of possible judicial interpretations and misinterpretations.

Trade Secrets and Computer Programs

Jurisprudential background. The law of trade secrets, unlike that of patent and copyright, is not a creature of statute. It is a common law doctrine which protects, in certain specified circumstances, against improper acquisition and/or disclosure of confidential business information. Although the obligation not to disclose trade secrets generally arises out of a contractual relationship, express or implied, between the proprietor of the information and the recipient, the applicability of the doctrine does not depend on the existence of a contract between the parties. In the words of Lord Denning, the protection afforded by the law of trade secrecy "depends on the broad principle of equity that he who has received information in confidence shall not take unfair advantage of it. He must not make use of it to the prejudice of him who gave it without obtaining his consent."[57] Before specifying those situations in which the courts have found an obligation of confidence to arise, an explanation of what type of subject matter is embraced within the term "trade secrets" will first be set out. The essential elements and scope of this form of protection will then be discussed.

The subject matter of trade secrets or confidential information has been defined judicially in extremely broad terms. Canadian courts have indicated that "trade secrets" cannot be defined precisely but they have, on numerous occasions, looked to the definition contained in section 757 of the U.S. Restatement of Torts:

> A trade secret may consist of any formula, pattern, device or compilation of information which is used in one's business, and which gives him an opportunity to obtain an advantage over competitors who do not know or use it. It may be a formula for a chemical compound, a process for manufacturing, treating or preserving materials,

---

57. Seager v. Copydex Ltd. ([1967] R.P.C. 349, pp. 367-368).

> a pattern for a machine or other device, or a list of customers.[58]

This definition would therefore encompass what is generally referred to as "know-how," a term used to describe valuable information acquired by a business enterprise, such as marketing and manufacturing techniques, organizational methods and technical data. In addition, non-technical information compiled by a business such as pricing formulae, supply sources, cost data, market studies, customer lists, promotional campaigns and future business plans could also be included within the ambit of trade secrets.

Patent law is ineffective in protecting much of the subject matter of trade secrets. In fact, most elements of know-how, such as organizational methods and customer lists, are unpatentable. Trade secret protection is the only available remedy for the unauthorized use of this type of proprietary information. Although formulae, manufacturing processes and machines or devices are patentable, there are several considerations which may make a firm choose to rely on trade secret protection as opposed to seeking a patent for its inventions. One such consideration is the disclosure requirement of patent law. If the patent is subsequently found to be invalid, the inventor not only loses his monopoly but will, in addition, have granted access to the technological information to his competitors. Another consideration is the time and expense needed to obtain a patent in the first place. Trade secret protection is immediate.[59] In addition, by not disclosing the trade secret in a patent application, the proprietor may avoid the use of the idea as a stepping stone by his competitors who may be able to devise an effective manner of avoiding the patent. Furthermore, patent protection is limited to 17 years, whereas trade secret protection is available for as long as the subject matter retains its confidential nature -- there is no maximum period of protection.

In order to succeed in an action for unauthorized disclosure or use of a trade secret, the plaintiff must establish three elements: that the information possessed the necessary quality of secrecy; that such secret information had been communicated to the defendant or to a third party from whom the defendant received it in circumstances giving rise to a duty of confidence; and that there was unauthorized use of the information to the detriment of the plaintiff.

(a) Secrecy. Only information which is confidential (in the sense that it is not a matter of general knowledge in the trade) will qualify for

---

58. Again, the emphasis has been added to demonstrate that the list is not exclusive of other trade secrets.

59. There are no initial legal expenses involved but the costs of maintaining safeguards to preserve secrecy can be substantial.

trade secret protection. Moreover, once the possessor of a secret voluntarily discloses it without restrictions, express or implied, the protection will be lost. In general, disclosure of the secret must be confined strictly to a limited group of persons, such as employees and potential licensees, in order for it to retain its confidential nature.

The unrestricted sale to the public of machines or other products which, when examined or disassembled, disclose the secret either through reverse engineering or mere visual inspection will generally destroy the confidential nature of the information.[60] In summary, the factors the courts have looked to in determining whether the information possessed the requisite degree of secrecy include:

(1) the extent to which the information is known outside the business;

(2) the extent to which it is known by employees and others involved in the business;

(3) the extent of the measures taken to guard the secrecy of the information;

(4) the value of the information to competitors;

(5) the amount of effort or money expended in developing the information; and

(6) the ease or difficulty with which the information could be acquired or duplicated by others.

(b) When a duty of confidence arises. The courts have held that the following relationships give rise to an obligation not to disclose confidential information:

(1) In the relationship of employer and employee, there is an implied term that the employee will not disclose the confidential information of his employer regardless of whether there is a written contract of employment. This obligation continues even after the employment is terminated. The employee may use his ordinary working knowledge, skill and general experience acquired in his earlier employment in any subsequent employment. The extent to which an employee may do so without breaching his obligation of confidence to his former employer is, however, an area of the law that is

---

60. O. Mustard & Son v. Allock & Co. Ltd. and Dosen ([1963] 3 All E.R. 416).

fraught with uncertainty. In the words of one commentator:

> The distinction between protectable trade secrets known by an employee on the one hand, and the employee's personal skill, knowledge and experience on the other, is often very fine; there is no rule of thumb. The employee's personal skill, knowledge and experience is not confined to what he picks up himself on the job, but may include information taught by his employer. Nor is the distinction safely made on the basis that any information the employee can carry away in his head is his personal knowledge, although it may be valid to conclude that if he has to copy it to carry it away, it is not part of his personal stock of information. And if he has to learn the information surreptitiously and not in the course of his duties, the knowledge gained is unlikely to be considered "personal to the employee." (Fogo, 1971, p. 19)

In general, the courts have been extremely reluctant to prohibit an employee from using his or her skills in subsequent employment; clear and convincing evidence of confidentiality and disclosure is required when a plaintiff alleges breach of confidence by a former employee.

(2) An obligation of confidence has also been held to arise if the owner of a trade secret has disclosed it to another enterprise for a specific purpose such as having a mould or dye made for future production.[61] The recipient of a trade secret in this circumstance must use the information solely for the purpose for which it was entrusted to him. Liability will attach if the recipient discloses it to others or makes use of it for his own purposes.

(3) The relationship between two parties in the process of negotiating for the licensed use by the recipient of a trade secret will give rise to an obligation of confidence even if no agreement on the use of the trade secret is ever entered into.[62]

(4) Finally, any relationship governed by an express written contract containing terms dealing with confidentiality and disclosure will

---

61. Saltman v. Campbell ([1948] 65 R.P.C. 283).

62. Coco v. Clark ([1969] R.P.C. 41).

give rise to the obligation. In these cases the problem is merely one of construing the contract.

(c) Position of unrelated third parties. Trade secret protection is a right exercisable against only specified persons who stand under an obligation of confidence to the owner, unlike patents and copyright which confer a broad property right exercisable against anyone.[63] It is now settled law that a third party who intentionally induces another to breach an obligation of confidence will himself be liable to the owner of the secret.[64] However, trade secret remedies are generally ineffective against unrelated third parties who receive or purchase confidential material in good faith and without notice that the party selling or communicating the secrets is under an obligation of confidence to another.[65]

Problems associated with trade secret protection of programs. There is no doubt that computer programs and associated documentation may qualify for trade secret protection. In fact, this branch of the law is currently the most widely adopted method of legal protection in the software industry (see Tables 18 and 19, Chapter III). Typically, employees of software firms sign contracts which prohibit them from divulging trade secrets and confidential information to outsiders. A software package is generally leased to customers under a restrictive licence agreement which contains terms preventing the customer from giving outsiders access to it and restricting the use to which the software may be put.

The major defects in trade secret protection, so far as computer programs are concerned, can briefly be summarized as follows. First, an employer who has hired a programmer to develop and write software will not be able to prevent that programmer from using the knowledge, skill and experience acquired in that job in any subsequent employment. If the programmer quits to work for a competitor and his initial employer suspects that he has divulged confidential information to the competing firm, the evidentiary problems of proving a breach of confidence are extremely onerous.[66] As mentioned earlier, the distinction between the utilization of an employee's personal skill, knowledge and experience on the one hand, and the disclosure of confidential informa-

---

63. Trade secrets is a right in personam rather than a right in rem. Consequently, its scope is not as broad as a cause of action based on property rights.

64. British Industrial Plastics v. Ferguson ([1941] 58 R.P.C. 1).

65. Fraser v. Evans ([1969] 1 Q.B. 349, p. 361).

66. For an example of some of the potential difficulties in this area, see Data General Corp. v. Digital Computer Controls, Inc. (297 A. 2d 433 [Del. Ch. 1971] aff'd, 297 A. 2d 437 [Del. 1972]).

tion on the other, is often extremely difficult to make. The tendency of the courts is to resolve any doubt in favour of the employee. Moreover, even if the employee has blatantly copied his former employer's software and disclosed it to a subsequent employer, no liability will ensue if the employee can show that his former employer did not take adequate safeguards to preserve the program's secrecy.

A second problem with trade secret protection is its inappropriateness with respect to general or specific-purpose programs having a potentially large market. Wide unrestricted sale and distribution destroys "secrecy," especially if the information sought to be protected can be discovered by visual engineering.[67] Third, the ongoing expense of maintaining program secrecy is often substantial. Even momentary laxity on the part of the owner may result in the loss of the confidential nature of the program. Finally, in most cases it will be difficult to prevent innocent third parties, outside the contractual or fiduciary relationship, from using a program disclosed to them without the authorization of the owner. These problems with trade secrecy protection for computer software will become increasingly important in the next decade as clients increasingly make use of package software rather than custom-developed or in-house software. One of the major advantages of copyright versus trade secrecy protection is that copyright can reduce the costs firms must incur to protect their intellectual property.

In the United States there exists some controversy over the legal issue of whether state trade secret law has been preempted by the federal Copyright Act of 1976. Section 301 of the federal Act states that: "Thereafter, no person is entitled to any such right or equivalent right in any such work under the common law or statutes of any State." The legislative history of the section indicates, however, that it was the intent of Congress only to abolish state common law copyright and not

---

67. WIPO states:

> At present the largest amount of expenditure on computer software seems to be devoted to the creation and maintenance of specific purpose user programs, not of general applicability; since such programs are not of direct interest to third parties, their misappropriation is relatively unlikely in view of the adaptation required. However, there is a trend toward the creation of computer programs that are of interest to more than one user or even of general and widespread utility and thus can help save expenditures; such a trend toward standardized user software is likely to increase as computers become more accessible to the public. (1978, p. 10)

to preempt state trade secret law.[68] Furthermore, in the Supreme Court decision in Kewanee Oil Co. v. Bicron (4.6 U.S. 470 [1974]), it was held that federal law does not generally render inoperative state trade secret law. Hence, the better view is that trade secret law in the United States has been unaffected by the federal Copyright Act of 1976, although this is not a settled question. In Canada, this constitutional problem does not arise with respect to provincial trade secret law because section 45 of the federal Copyright Act provides that the statute does not affect "any right or jurisdiction to restrain a breach of trust or confidence."

---

68. See also 17 U.S.C., s. 117 (1976) which contains a "freezing provision."

Chapter III

POLICY ALTERNATIVES AND RECOMMENDATIONS

The first two chapters of this study described the computer software industry, along with the history of the legal problems of defining and protecting the intellectual property value of its output. This chapter examines each of the policy alternatives and offers suggestions concerning an analysis of the expected costs and benefits of each alternative. As Demsetz (1969) has so clearly emphasized in his scathing criticism of Arrow, discussions of policy alternatives must keep in mind feasible standards for comparison. Thus, each alternative will be compared with the status quo, which is based primarily on the use of market, technological and trade secret protection, rather than with some idealized perfectly competitive norm. Later in the discussion, additional alternatives will be included in the comparisons. It will be argued that some weak form of intellectual property protection is desirable for computer software. Estimates are presented indicating that the net benefits of copyright protection for computer software are positive, but these estimates are qualified by the recognition that they may be too high when international factors are taken into consideration and too low when the future markets for firmware are considered.

Need for Additional Protection?

The computer software industry has been growing dramatically during the past decade. Chapter I chronicled both the rate and the reasons for its growth. It is reasonable to ask whether there is any need to protect software beyond the protection currently available technologically and through rapid marketing or trade secret protection. If the industry has grown rapidly and offered sufficient incentives for entry and technological development, it can be argued that there is no justifiable basis for creating even a limited property right and monopoly in specific computer software. The crux of this argument is that providing intellectual property protection in addition to what is now available would provide no further benefits for society and that allowing even a limited monopoly right would generate some extra costs for society.

In an exploration of the need for intellectual property protection of computer software, Miller surveyed industry members to determine what they believed to be effective means of protection (1974 and 1978). The firms were asked to indicate the degree of effectiveness of alternative means of protecting their software. Their responses are indicated in Table 18. As can readily be seen, most firms found a lease with a confidential disclosure clause or trade secret licensing to be the most effective modes of legal protection. Interestingly, 58 per cent of the respondents claimed to have tried copyright protection, yet there had been fewer than 1000 copyright registrations of computer software in the United States when the survey was conducted. It is peculiar that respondents would claim that copyright protection had any

Table 18

Preferred Modes of Legal Protection
(figures indicate percentage of
respondents answering in each category)

Mode of protection	Degree of Effectiveness: not at all effective	somewhat effective	very effective	completely effective	not used
Lease with a confidential disclosure	.03	.23	.35	.16	.23
Trade secret licence	.13	.16	.26	.10	.35
Copyright	.09	.26	.16	.07	.42
Physically limiting access to technology	.07	.16	.20	.13	.44
Cryptographic coding	.13	.10	.07	0	.70
Other:					
software lock	0	0	.08	0	.97
controlled support	0	0	.03	0	.97
patent	.03	0	0	0	.97

SOURCE: Miller, 1974, p. 56

effectiveness at all in the United States prior to 1978 in view of the fact that the U.S. copyright law did not clearly include software and that the only legal case involving copyright and software did not directly support the copyrightability of software. One possibility is that software producers found that copyright protection of their instruction manuals provided some safeguard for their software. Or perhaps the respondents expected that copyright protection, if available and if supported by the courts, would have some modicum of effectiveness.

In a later study carried out for CONTU, Miller surveyed a much larger sample of firms using a somewhat more extensive questionnaire (1977). As with the 1973 survey, the 1977 survey was accompanied by a letter explaining that the major interest in the survey concerned the possible effects of copyright protection. The results, presented in Table 19, indicate a decreased use of copyright between 1973 and 1977 as the number of firms surveyed quadrupled. Interestingly, of those firms which used copyright protection, the views concerning its effectiveness became more polarized with many more finding it ineffective and as well, more finding it to be effective. A plausible explanation of this result is that as firms attempted to protect their software with copyright some were quite disillusioned because the copyright protection was ineffective against imitators' uses of an algorithm. The less sophisticated firms may have been attempting to use copyright to protect their algorithms and have been disappointed; the more sophisticated firms have made extensive use of trade secrecy tactics (e.g., encryption and release of only the object program) to protect their algorithms, but they know to use copyright to protect only a particular expression of an algorithm. Miller's explanation of these results is similar to the one presented above: "We explain these developments by observing that as copyright usage has increased, more respondents have used this relatively inexpensive, accessible mechanism to try to protect more programs that are easily designed around (1977, p. 20).

Morgan's comment on Miller's result is:

> This conclusion should have come as no surprise to anyone in this country, since what it means is that:
>
> (i) in practice, infringement of title is impossible to detect or stop (e.g. a software house adapts programs written for client A for use by client B); or
> (ii) title can be protected physically or technologically (e.g. delivery to the licensee only, of an object code version of the program suitable for running on his machine, and withholding all source language versions);
> or
> (iii) title can be protected by contract. (Morgan, 1979, p. 62)

Table 19

Preferred Mode of Protection
(figures indicate percentage of
respondents answering in each category)

Mode of Protection	Degree of Effectiveness: not at all effective	rarely effective	somewhat effective	fairly effective	very effective	completely effective	Frequency of use*
Patent	.82	0	0	.18	0	0	.04
Copyright	.55	0	.05	.1	.15	.15	.2
Trade secret	.29	.05	.14	.24	.14	.14	.21
Release of object program only	.17	0	.04	.08	.33	.38	.3
Know-how requirement	.28	.17	0	.17	.17	.22	.13
Cryptographic coding	.5	0	.17	.25	.08	0	.4
Other means of limiting access	.27	0	0	.13	.06	.53	.17

SOURCE: Miller, 1977, p. 19

*The figures in this column relate to the entire sample.

In other words, Morgan is asserting that copyright protection offers firms little, if anything, more than is presently available (i.e., it offers no private benefits and hence no incentive for the private sector to generate any social benefits).

If copyright protection of software offers no private benefits, then it follows that it also imposes no social costs (other than legal and administrative costs). The private benefits of the monopoly due to copyright exist only if the monopoly power can be and is exercised. If these private benefits are small, then the deadweight losses due to the lost consumers' surpluses are likely also to be small and concern about the creation of monopoly distortions is a gross exaggeration. This is an important point deserving reiteration. Critics of intellectual property protection of computer software (or anything else, for that matter) cannot have it both ways. They cannot argue that it has no benefits and yet that it would impose immense social costs through resource misallocation. The only way legal intellectual property protection can have no social benefits is if it creates no incentives for people to generate intellectual output (i.e., if it creates no private benefits). And if it creates no private benefits and no incentives for resource reallocation, it cannot then be criticized for creating resource misallocation.[1] As was pointed out in Chapter II, the concern of most economists is how much short-run reallocation of resources is desirable in order to attain longer-run goals. In the absence of private benefits, this concern poses no dilemma whatsoever; if there is no short-run reallocation, there can be no short-run misallocation arising from copyright or similar legal intellectual property protection.

Second, the validity of Morgan's (implied) conclusion that the social benefits of copyright or similar protection for computer software would be zero is questionable. Further on in Miller's first study, the following statement appears:

> Eighty-seven percent of all respondents could not think of a single instance in which computer programs representing a significant level of innovation were not developed or marketed because of inadequate protection. The companies that thought the law had been a barrier cited examples in which fear of easy plagiarism or unautho-

---

1. Several qualifications to this rather strong statement are important: (1) rent-seeking behaviour by potential competitors may erode the private benefits and yet create a social deadweight loss; (2) it is conceivable that externalities could be generated which would reduce private benefits and leave a substantial social loss; (3) the analysis uses consumers' surplus, a concept which, though often used in analysis of this type, is not always applicable; and (4) the analysis avoids the general problem of "second best," (i.e., it examines only one industry in the economy without considering feedback effects from one industry to another.

> rized disclosure might prevent recoupment of development costs. The situations cited involved such techniques as paging programs for virtual memory computers, an innnovative approach to developing multiprogramming capability on the IBM 360/20, and systems software for organizing computer program libraries. (Miller, 1974, p. 59)[2]

Despite whatever biases may have existed in his survey due to its small size and its use of only ADAPSO members, Miller's results indicate that a non-negligible number of firms were aware of instances in which technological development had been impeded for fear that the lack of intellectual property protection might give rise to misappropriation.

In his 1977 study, Miller found that "seventy-four percent of the sample had never rejected or abandoned a program because of the presence or absence of protection and 65 percent would not change their marketing even if protection were provided" (p. 25). "Seventy-seven percent of the sample knew of no instance of aborted marketing or development" (p. 27 -- emphasis in the original). It certainly appears from the results of Miller's two studies that an increasing number of instances was occurring in which computer software was not developed or marketed as a result of the lack of protection. It is unfortunate that Miller did not specify in this question what he meant by legal protection but, as indicated earlier, his cover letters, along with the other questions in the surveys, indicate that the surveys were being carried out to study copyright protection. It can therefore be assumed that most firms answering these questions would have interpreted "legal protection" as meaning copyright protection.

One of the reasons that fewer firms were unaware of impeded software development and marketing was that proprietary software packages were becoming increasingly important in the software industry during the mid-1970s (see Chapter I). The projection by Business Week that proprietary software packages will take on an even greater role in the industry suggests that the lack of legal protection for these packages will increasingly impede development in the industry in the future (Figure 2).

Miller's more detailed results support this projection that copyright will become more important in the future. He segmented his sample according to the type of software produced by each firm and asked the firms to indicate the significance of protection for their product(s) (Table 20). Miller summarizes the results by saying:

---

2. Miller's survey has a wide margin for error. It was quite small in scope and covered primarily larger firms, which might be expected to have the most to gain from intellectual property protection of software.

> With allowances for a difference in the way in which the data is expressed, the CONTU results are strikingly similar to those of the 1973...survey. What they show is that the more universal and widely marketed the program the more important is protection. This is a characteristic of general business programs, which can be addressed to such functions as payroll and receivables anywhere, and also of systems software in which a program can be used for a particular computer in a variety of installations. On the other hand, the more technical and unique the program the less significant protection appears to be. This finding is consistent with information which was supplied to us about the programs which were being marketed. (Miller, 1977, p. 22)

And, in response to an open-ended question, "The comment most often repeated was that an apparent lack of interest in legal protection was related to the fact that they did not market proprietary software" (1977, p. 28).

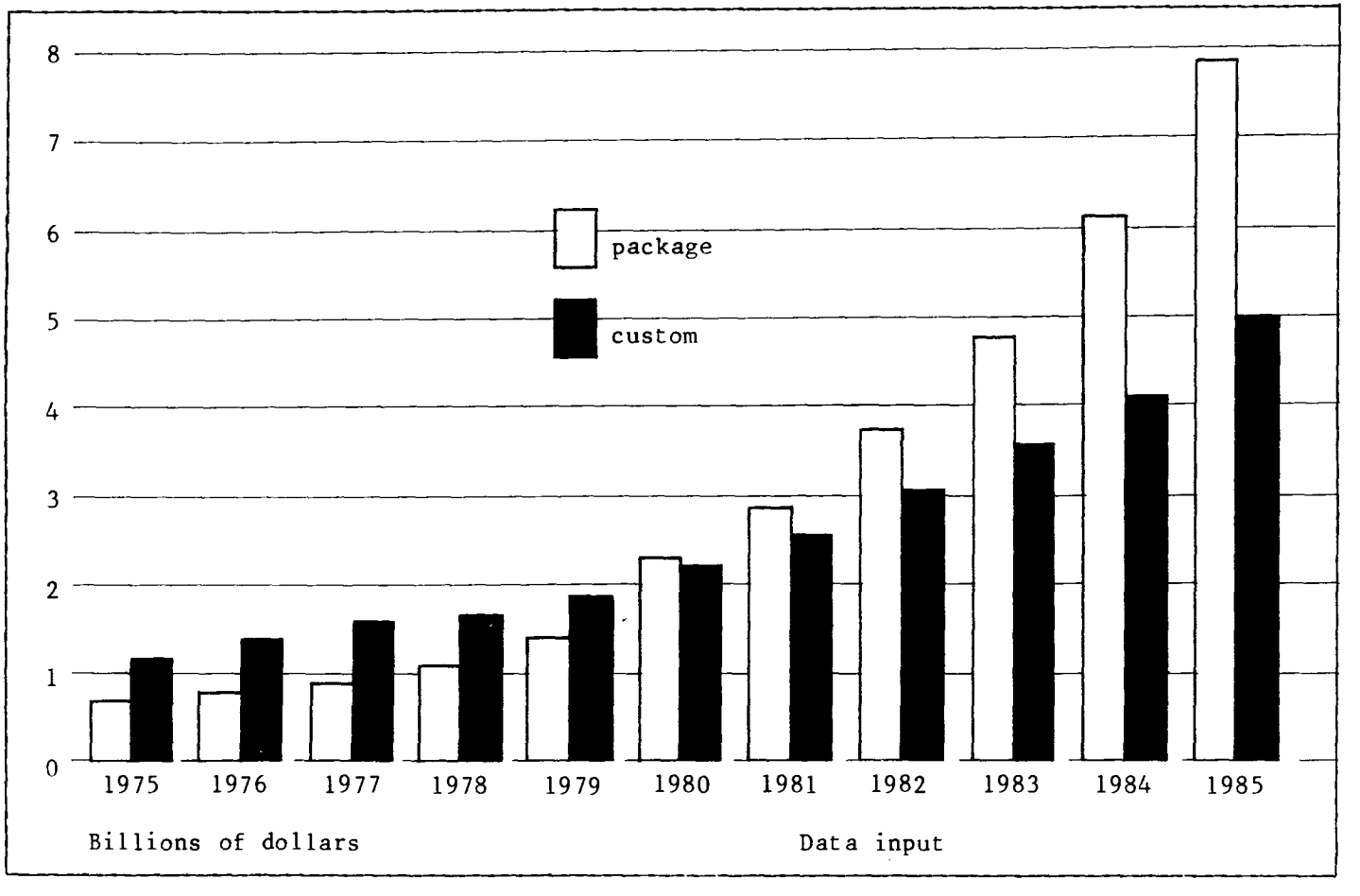
In conclusion, the software industry has grown rapidly in the past decade, even in the absence of explicit legal protection for its intellectual property. An important question for policy prescription, however, is how fast would the industry have grown had better protection been available? If the industry could have grown even faster with intellectual property protection, there is a reasonable chance that protection would have had some net social benefits. A related question addresses the future trends in the industry. It has been seen that industry experts expect that there will be even more emphasis in the future on proprietary software packages, precisely those which Miller found to be more amenable to explicit legal protection because these programs require wide-scale marketing and are hence more costly to protect using trade secrets. What type of legal protection will be the most valuable to society for firms in this segment of the industry is vague, although it is reasonable to assume that Miller and his respondents had copyright protection in mind. The indication from this paper's industry study and from Miller's studies is that the continued growth of the proprietary software package segment of the computer software industry has made and will continue to make the potential net social benefits from legal, possibly copyright, protection of computer software increase over time.[3]

---

3. This preliminary discussion has not mentioned computer firmware, a segment of the industry to be considered in detail later in this chapter. At this point all that need be noted is that the problems of traditional trade secret protection for products as general and marketed as widely as firmware are possibly even greater than for proprietary software packages.

Figure 2

How Sales of Standard Packages Will Overtake Custom Software



SOURCE: Business Week, September 1, 1980, p. 56.

Table 20

Significance of Software Protection by Function*

	No significance	Some significance	Great significance
(a) General business and financial applications (e.g., accounting, inventory control, payroll)	.17	.33	.5
(b) Business planning operations (e.g., planning models, simulations, operations research)	.5	.24	.26
(c) Complex production/distribution control operations (e.g., linear programming)	.55	.19	.26
(d) Engineering and scientific applications	.53	.19	.28
(e) Data and statistical analysis	.51	.17	.31
(f) Project management and control	.54	.23	.23
(g) Systems software (e.g., compilers, monitors, new techniques for more efficient machine utilization)	.4	.17	.43

SOURCE: Miller, 1977, p. 22.

*Figures indicate percentage of respondents answering in each category.

Framework for Analysis

Before examining in greater detail the net social benefits of various types of legal protection for computer software, this section sets out a theoretical structure for the analysis. This theoretical structure is quite simple and is commonly used for cost-benefit analyses of government policies. It is based on the fundamental economic tools of supply and demand analysis.

Consider the market for computer software. It is assumed that there is a downward-sloping demand curve for software; in other words, if the price of software were to decline, the amount of software demanded would increase. Treating all software as part of the same market is a type of aggregation often made in economic analysis. A software package designed to carry out some esoteric engineering calculation is not the same as a software package designed for general business accounting; nevertheless, all software is included in one industry for this analysis, just as all products in broad industry classifications are lumped together for most industry studies.

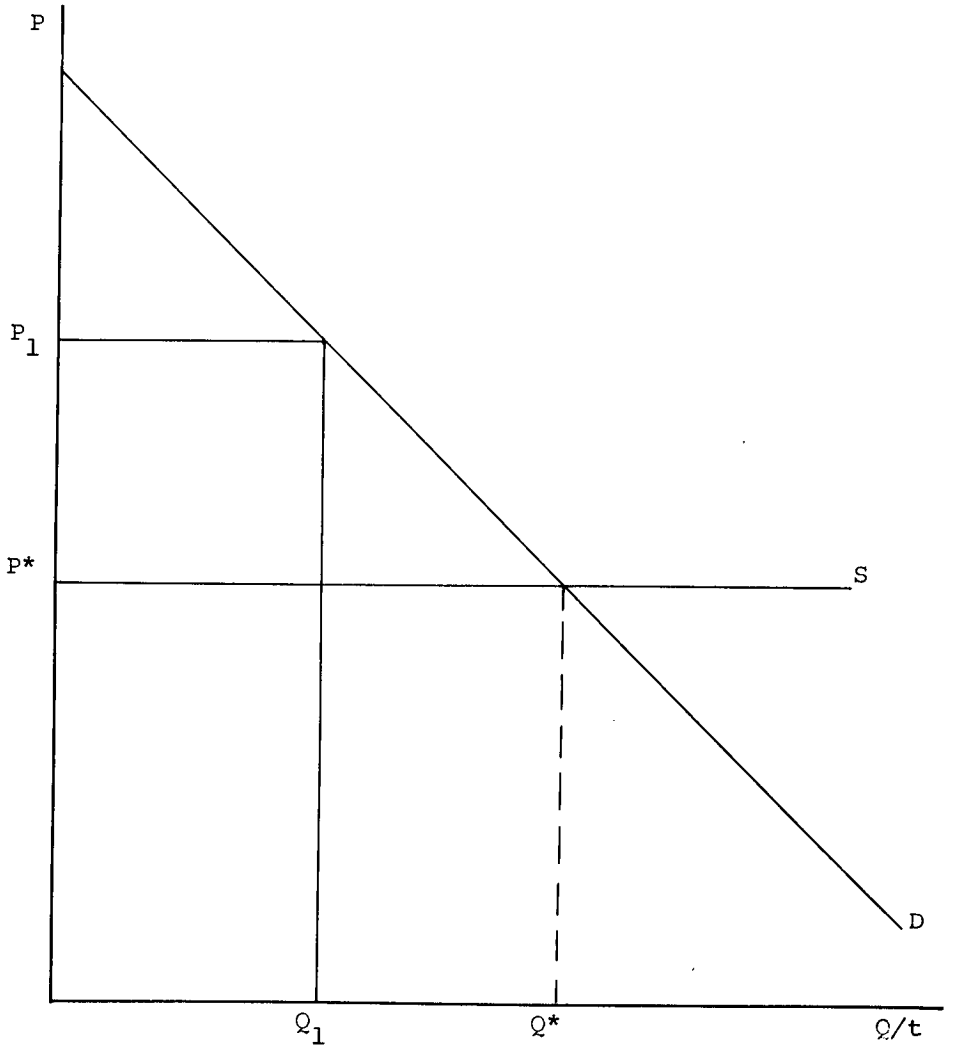
The long-run supply curve for computer software is assumed to be horizontal. The reason for this assumption follows, in part, from the finding in Chapter I that the industry is reasonably competitive and has low barriers to entry. These industry conditions imply (assuming input prices remain unchanged) that if the price ever exceeded the average costs of developing, producing and marketing software, other firms would enter the industry and, through competition, drive the price back down to the average costs (including a normal rate of return). The assumed shapes of the demand and supply curves are shown in Figure 3, along with the equilibrium quantity of software sold, Q^*, and the equilibrium price, P^*, which is also equal to the long-run average costs of producing computer software.

It is important to see in Figure 3 that, except for the Q^{*th} unit of computer software, all of the other transactions could have taken place at higher prices in the sense that purchasers would have been willing to pay more than P^* for the software. For example, Q_1 units of software could have been sold at P_1 but, because of competition, the price was driven down to P^*. In a sense, then, for every unit which could have been sold at a higher price but which was sold at P^*, a surplus has been generated for the purchasers equal to the difference between P^* and what they would have been willing to pay. This surplus is shown as the shaded triangle PP^*G in Figure 4.

One of the reasons that intellectual property protection can be beneficial to society is that it reduces the average costs of producing computer software by decreasing the costs of protecting against misappropriation by imitators. As has been seen, many computer programs would continue to be produced, possibly without much change in the costs of production and marketing, were intellectual property protection available. Nevertheless, the major reason that programs were not produced which might have been produced with explicit legal protection

Figure 3

The Market for Software



was that the costs of private protection against misappropriation were expected to be greater than the expected gains from producing the software. Furthermore, some programs currently being produced and marketed could be produced and marketed at lower costs were legal protection available.[4] As a result, on average the provision of more explicit legal intellectual property protection would reduce the average costs of producing software and in Figure 4 would shift the supply curve downward to S'. This reduction in costs would generate additional social benefits (ignoring, for the moment, legal and administrative costs) equal to the area of the shaded trapezoid in Figure 4, P*GFP'. This area, B, is equal to

$$B = \frac{1}{2} (P^* - P') (Q^* + Q') \qquad \text{(Equation 1)}$$

If $(P^* - P')$, also to be represented by ΔP, cannot be determined directly and if the price elasticity of demand is known, ΔP can be determined from the equation for elasticity:

$$E = \frac{\Delta Q}{Q} \cdot \frac{P}{\Delta P} \text{ or} \qquad \text{(Equation 2)}$$

$$\Delta P = \frac{\Delta Q}{Q} \cdot \frac{P}{E} \qquad \text{(Equation 3)}$$

where ΔQ is $Q' - Q^*$. Substituting equation (3) into equation (1), results in:

$$B = \frac{1}{2} \frac{\Delta Q}{Q} \cdot \frac{P}{E} (Q^* + Q'). \qquad \text{(Equation 4)}$$

Because $Q' = Q^* + \Delta Q$, equation (4) can be rewritten as:

$$B = \frac{1}{2} \frac{\Delta Q}{Q^*} \cdot \frac{P^*}{E} (2Q^* + \Delta Q), \text{ or as} \qquad \text{(Equation 5)}$$

$$B = \frac{\Delta Q P^*}{E} + \frac{(\Delta Q)^2 P^*}{2Q^* E} \qquad \text{(Equation 6)}$$

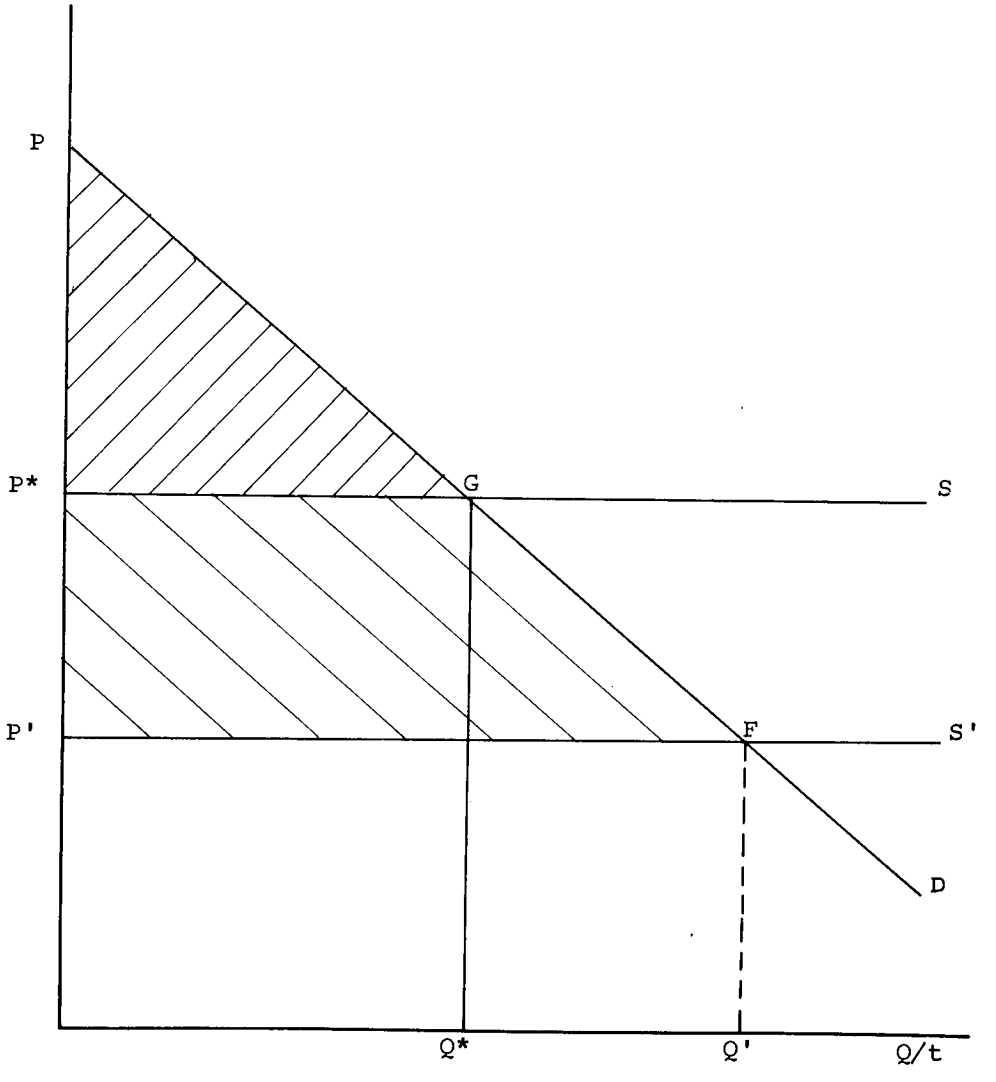
Examination of equation (6) reveals that, if P* and ΔQ are understated and E and Q* are overstated, the estimates of the social benefits of protection will be biased downwards. This procedure is followed here because the estimate of the values of each of the variables is quite

---

4. Of course, copyright would also create a limited monopoly right which could tend to inhibit a downward movement of prices. As emphasized throughout this study, this monopoly right would be extremely limited, protecting only a unique expression of an algorithm; others would be free to use the same algorithms in their own independently created expressions. See pp. 130-131.

Figure 4

The Social Savings Due to Intellectual Property Laws



imprecise. If the deck is stacked against finding significant social benefits and some are still found, there is more reason to be confident of the existence of these benefits. Later in this chapter, possible maximum values for B are also explored by biasing the estimate upwards. An alternative explanation of the estimation method is presented in Appendix D.

Before using this framework to analyze policy options, however, some of the potential problems in using it should be pointed out.

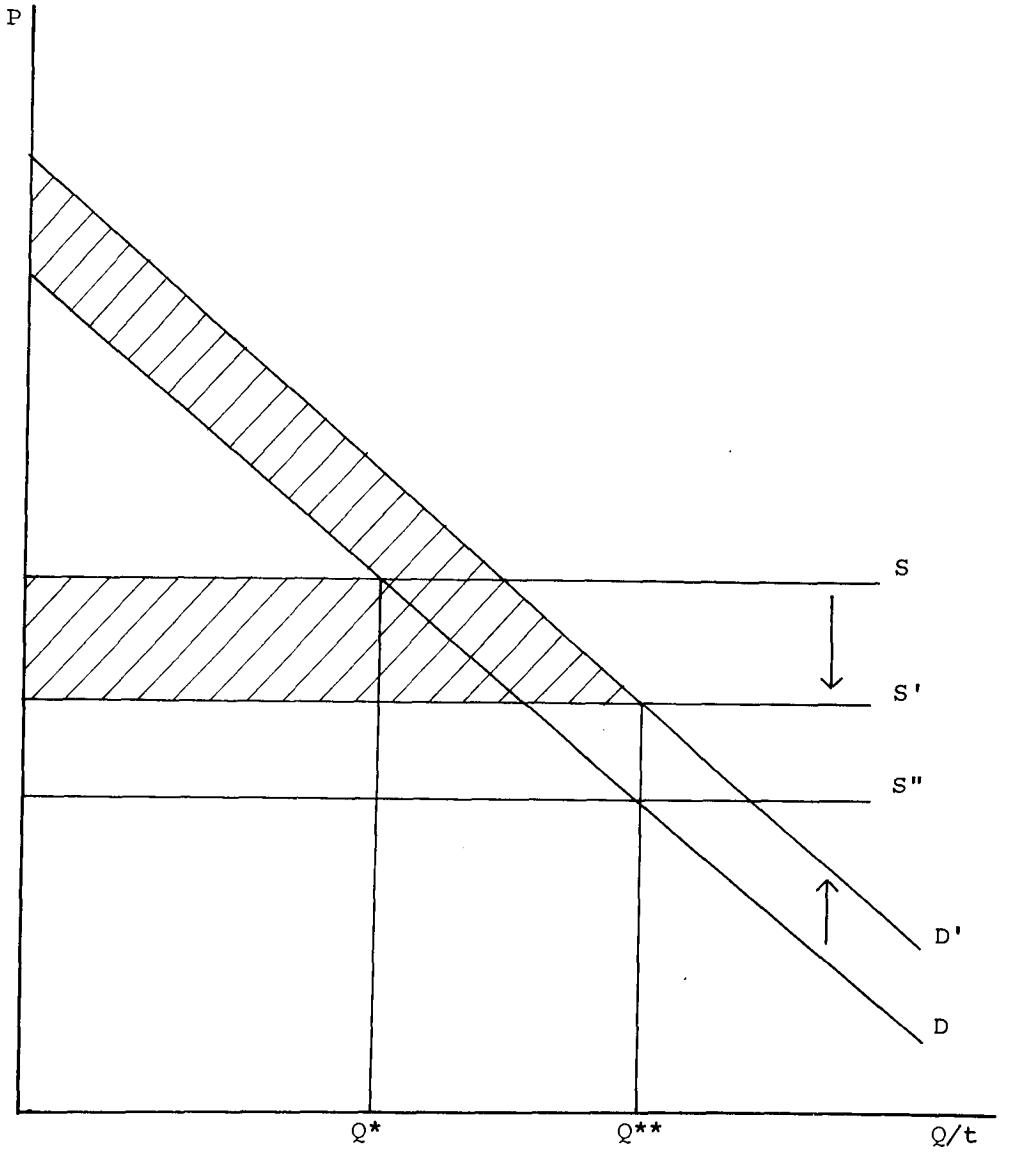
Non-horizontal supply curves. First, the assumption of a horizontal supply curve may be inappropriate. Use of a horizontal supply curve has been proposed on the basis of long-run economic analysis that entry will take place and that, even in the computer programming professions, mobility will occur, competing the long-run price of software back down to average costs. Because intellectual property law is concerned with long-run incentive effects, it seems appropriate to use a long-run economic model as the framework for analysis.

Even in the long run, though, the supply curve for computer software may not necessarily be horizontal. If there are some inputs for the production of software which would be scarce even in the long run, even allowing for factor mobility, then the costs of producing computer software would increase as a result of entry and attempts within the industry to increase output. In this case, the supply curve would be upward sloping and the social benefits of any given cost reduction for developing and marketing software could be smaller than those estimated using equation (6). This caveat must be kept in mind when the more specific uses of equation (6) are explored later in this chapter.

Demand curve shifts. Estimating social benefits using equation (6) assumes that intellectual property protection would have no effect on the demand for computer software. This assumption may not be an accurate reflection of reality. Intellectual property protection could increase the demand for computer software in two ways. First, by reducing information costs it will make any particular package more valuable to potential clients. These clients probably view the costs of a software package as including the price of the package as well as their search costs in finding out what is available and from whom. To the extent that intellectual property protection of computer software reduces these search costs, the area of the trapezoid will be reduced, and equation (6) will understate the social benefits of the protection. Diagrammatically, this effect is shown in Figure 5 as the entire shaded area due to both the cost reduction and the increase in demand. Practically, however, the effect may be unimportant if the movement from Q^* to Q^{**} is assumed to be due solely to cost reduction. In this case, for a given price elasticity of demand, the supply curve would be assumed to shift downward, not to S' but to S'', and a larger trapezoid would be estimated. Although there is no reason to expect this larger trapezoid to be equal in area to the shaded area shown in the figure, it will generally be a reasonably close approximation. In addition, if intellectual property protection increases the average quality of software, it will also, on

Figure 5

The Social Savings from Intellectual Property Laws
When Both Supply and Demand Are Affected



average, increase the demand for software and could increase social benefits in this way as well, depending on thecosts of producing higher quality software. Both of these effects could prove to be important in the long run but are not included in the estimate of social benefits which is provided by use of equation (6).

Patent Protection

Despite continued U.S. Supreme Court holdings that specific software was unpatentable, there is continued and considerable interest in the patentability of software. One reason for this interest is that the court has continually denied any implication from its decisions that all software is unpatentable (see Chapter II). In Canada, the Patent Appeal Board has reversed itself and now holds software to be unpatentable, yet this reversal has been appealed to the federal courts and numerous other cases before the Patent Appeal Board are pending. Evidence of continued interest in the patentability of software comes from the fact that by 1977 only 1205 computer programs had been registered for copyright protection in the United States.[5] At the same time, Gemignani estimates that approximately 450 patent applications for computer software are filed each year (1980, p. 304). And a recent article by Novick and Wallenstein argues strenuously for the patentability of algorithms (1980).

The major reason for this continued interest in patent protection for computer software is that it would provide considerable private benefits to software developers by allowing them monopoly power over an algorithm, regardless of how that algorithm is expressed. It permits the patentee to preempt the algorithm, allowing protection against innocent infringers and independent discoverers of the algorithm. The expected monopoly gains from such protection would be considerably greater than for copyright protection, which grants a limited monopoly over only the unique expression of an algorithm but does not afford any protection to the algorithm itself. Because the potential monopoly gains are great, developers are willing to devote considerable resources to attempting to achieve them even in the face of a low probability of success. Unfortunately for society, most of the resources devoted to seeking patented software are currently legal and political. If they were technological, increasing research efforts toward finding new, patentable software, the arguments in favour of software patentability would be stronger in that such resources would aid in promoting economic growth, the primary rationale for intellectual property protection. If patent protection for computer software were clearly available, of course, resources would be shifted from the legal and political arenas to the research and development arena and technological growth might well be fostered. Nevertheless, patent protection might just as easily impede economic growth by allowing the preemption of

---

5. Recall that registration is required for copyright protection in the United States.

mathematical and scientific principles and by defeating one of the purposes of patent protection -- the dissemination and widespread availability of ideas.

It was argued at the beginning of this chapter that there can be no misallocation of resources unless there is also a reallocation caused by a particular policy. What was demonstrated in the preceding paragraph is that patent protection provides an incentive for the reallocation of resources and hence creates the possibility for a misallocation of resources as well.

The patentability of computer software can be rejected on three grounds: (1) it is inconsistent with the traditionally accepted ideas of patent law; (2) it would impose social costs through preemption of ideas; and (3) it would have great administrative costs.

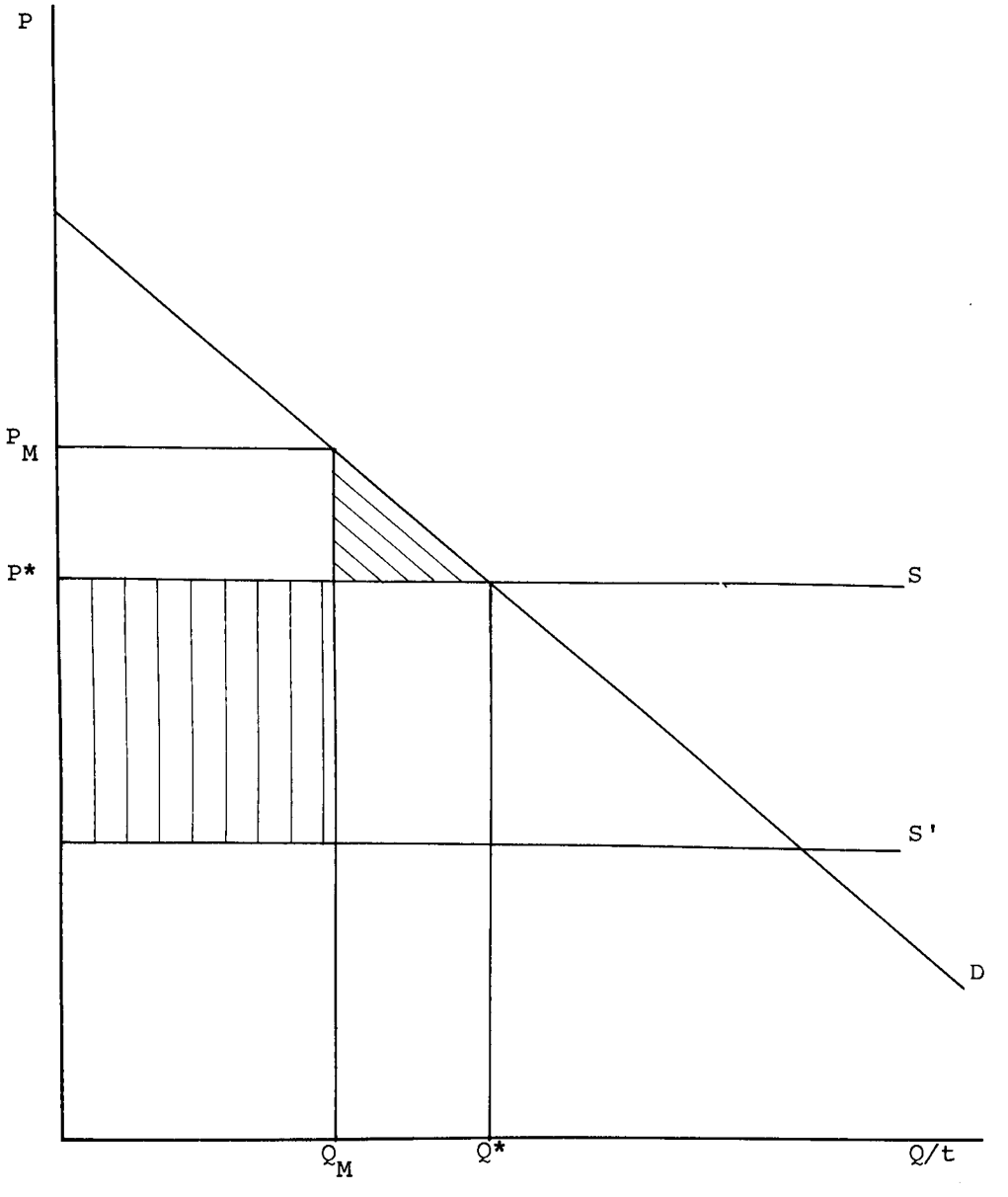
Inconsistency with traditional patent law. Both Canadian and U.S. courts have long held that their respective legislation clearly requires novelty and unobviousness as two of the criteria for patentability. New computer programs would generally have no difficulty satisfying the former but would have considerable difficulty satisfying the latter criterion. The primary challenge to most patents or patent applications would be that the patentee or applicant did nothing more than anyone skilled in the prior art could do. Denying such challenges by allowing software to qualify for patent protection would represent a fundamental change in the interpretation of patent law and could have serious implications for patent law application for other products. Essentially, the argument here is that the recent patent decisions in both Canada and the United States have been legally correct and in keeping with the traditional rationales for patent protection set out in Chapter II.

Preemption of principles. In cases involving software patentability, the courts have made broad use of what has come to be known as the mental steps doctrine. Their position is that if the software simply enables a computer to do that which a reasonably intelligent human could do with a pencil and paper (ignoring the time involved) following a series of mental steps, then the software is unpatentable because otherwise it would grant a monopoly to a series of mental steps (i.e., a scientific or mathematical principle). Such principles are denied patentability on the grounds that society will be better served if such principles are freely available to all who wish to use them.

This rationale can be seen more clearly by using the framework for analysis set out in the previous section and with reference to Figure 6. Suppose, as in the last section, that allowing patent protection for software would reduce the costs of producing and marketing

Figure 6

Efficiency Gains vs. Consumer Losses



software so that the supply curve shifted downward from S to S'.[6] There would then be a social benefit, as before, resulting from the reduced cost of producing this software. In the case of patents, however, each type of software would have a preemptive monopoly over a class of computer software. Consequently, entry and competition would not drive the price downward toward the long-run average costs of producing software. Instead, each patented proprietary software package could be sold at a monopoly price, raising the average price of software and reducing output in the industry. This effect is indicated with P_M, the monopoly price, and Q_M, the monopoly rate of output. Relative to the status quo (essentially trade secrecy), it appears at first blush that society has gained an amount equal to the area of the shaded rectangle (these gains accrue to the software producers) and lost benefits equal to the shaded triangle, which represents what economists have come to refer to as the deadweight loss due to monopoly. Whether the rectangle is larger than the triangle depends on the price elasticity of demand, the strength of the monopoly power created and the amount of the cost reduction.

More careful thought has led economists to question this analysis, however. They point out that what appear to be social gains will probably be eroded as competing firms devote an "excess" amount of resources to the acquisitions of private gains. In this respect, then, the social benefits cannot be assumed to be equal to the area of the shaded rectangle and, in the extreme case, are completely dissipated by "rent-seeking" behaviour.[7]

In considering the possible patentability of computer software, other social costs must also be considered. The courts have consistently expressed a fear that granting such patents would preempt a theorem or principle not only as it is applied in the software industry but for any application. Their concern, which seems to be a valid one, in reference to Figure 6, is that the deadweight loss triangle is a serious understatement of the social costs because allowing patentability of theorems and principles would create monopoly deadweight losses in many industries in addition to the computer software industry.

Novick and Wallenstein have criticized the preemption basis for the rejection of software patentability. They argue that there are numerous software algorithms for applying the same mathematical principle and that patenting any one of the algorithms would not preempt the principle itself. To demonstrate their argument, they present an example of two computer programmers, Joe Scientist and Joe Engineer,

---

6. It is not precise in economic theory to refer to supply curves in a monopolized industry. In this case, the more technically-minded reader can view what are called supply curves here as long-run marginal cost curves.

7. See, for example, Posner, 1975, p. 807.

who have devised algorithms for making calculations according to the Pythagorean Theorem:[8]

> Scientist writes an algorithm that proceeds as follows:
>
> (1) take number (a) and multiply it by itself;
> (2) repeat step (1) with number (b);
> (3) add the results of steps (1) and (2) together (d^2);
> (4) guess at the number (c) whose value when squared will equal the previously obtained total (d^2);
> (5) multiply this guess (c) by itself (c^2);
> (6) compare (c^2) with (d^2);
> (7) if (c^2) does not equal (d^2), then return to step (4); and
> (8) if (c^2) equals (d^2), then relay the number (c) to the operator as the desired result.
>
> Engineer writes an algorithm that proceeds as follows:
>
> (1) take number (a) and look up in a supplied table the result of multiplying it by itself;
> (2) repeat step (1) with number (b);
> (3) add the results of steps (1) and (2) together (d^2);
> (4) look up in a supplied table what number (c) when multiplied by itself equals (d^2); and
> (5) relay the number (c) to the operator as the desired result.
>
> After writing their respective algorithms in an appropriate computer language, both Scientist and Engineer apply for patents on their work. Neither computer program would satisfy the criterion of patentable subject matter under the Supreme Court's current construction of the Patent Act. Why? Because each program "would wholly pre-empt the mathematical formula and in practical effect would be a patent on the algorithm itself!" These contentions are clearly preposterous. As to the latter contention, the algorithm also known as computer program is the precise subject of the patent claim. As to the former contention, neither claim can be said to "wholly pre-empt the mathematical formula" since, a fortiori, two competing and distinct claims exist! (Novick and Wallenstein, 1980, pp. 337-338)

---

8. The Pythagorean Theorem states that for a right triangle, the squared distance of the hypotenuse (the side opposite the 90-degree angle) equals the sum of the squared distances of the remaining two sides.

Novick and Wallenstein argue that patent protection should be granted because neither algorithm preempts use of the Pythagorean Theorem. The question that can then be asked is, what, then, is being patented? In the case of Joe Scientist, is it the computer application of an iterative process? In the case of Joe Engineer, is it the use of square and square root tables by a computer? Both algorithms involve fundamental scientific principles and would preempt their use in solving any other arithmetic problem. Perhaps what Novick and Wallenstein would patent is the combination of fundamental principles: the use of an iterative technique or the use of tables in conjunction with the Pythagorean Theorem. Even in this simple example, the algorithms are not subject matter suitable for patentability because solving the Pythagorean equation would generally be done in such a fashion, regardless of the use of a computer, and basic principles would be granted a monopoly inappropriately. More complex algorithms would involve the conjunction of a longer series of basic principles and the granting of patent protection would be less likely to preempt any single principle in these situations; nevertheless the courts' applications of the mental steps doctrine seems justifiable. And even if such complex algorithms were suitable subject matter for patent consideration, they should not be granted patent protection on the grounds of obviousness; a computer programmer skilled in the prior art could write such programs.

Administrative costs. In 1979, the Canadian Patent Office received approximately 23,000 patent applications and had a budget of between eight and nine million dollars, meaning that the average cost to taxpayers was approximately $350 per application.[9] If anything, the incremental costs per application for computer software would be greater than this average figure. Computer programs can be extremely long and complex, necessitating long hours of skilled examination and comparison with prior art by experts before deciding whether or not they would merit patent protection. The current delay from application to decision averages slightly more than two years and it would probably increase for complex software. These government costs do not represent all of the social administrative costs of patents, however. In addition, firms spend, on average, more than $1000 per application for legal fees simply to file a patent application (Table 17). Additional costs would be imposed on society by the lack of institutional competence within the judicial system to decide between competing arguments concerning alleged infringement of complex and lengthy algorithms (Gemignani, 1980, p. 301). In comparison with technological or trade secret protection of algorithms, these administrative costs may be low; trade secret protection requires that reasonable steps be taken to safeguard a secret and these reasonable steps could easily amount to more than $2,000 or $3,000 per program while technological protection requires the development of additional coding or technology, which could also utilize resources amounting to more than either the average or the incremental

9. Data provided by André Gariépy, Commissioner of Patents.

costs of patent protection.[10] In fact, it is this very saving in administrative costs of patent protection over trade secrecy and technological protection which adds to its desirability in its traditional coverage.

In conclusion, patent protection for computer software has some advantages over trade secret and technological protection. Its administrative costs may be lower and it may encourage broader dissemination of ideas. It is not at all clear, however, that the extended protection offered by patent over trade secret or copyright protection is socially desirable. According patent protection to software would deviate seriously from traditional interpretations of patent law in ways that might require a reinterpretation of the statute which would be unintended. Furthermore, monopoly rights should not be granted which would allow the preemption of scientific or mathematical principles. Although trade secrecy also grants a type of monopoly over basic principles, it does so only so long as the secret is not discovered. Trade secrecy does not preempt independent discovery and, as a result, allows for considerable incentive to utilize basic principles without the fear of an infringement suit. Consequently, it is recommended that algorithms not be awarded patent protection but rather be subject only to the status quo methods of protection: trade secrecy, rapid marketing and technological secrecy.

Industrial Design Protection

There do not seem to be any serious academic or political suggestions that industrial design protection be awarded to computer software.[11] Informal discussions with lawyers have always led to this possibility, however, for two reasons. The first is that the short term of protection is desirable to many people who believe that even a 50-year term of protection would be longer than necessary to provide appropriate incentives for the development and marketing of additional software. The second is that firmware has been raised as a possible candidate for industrial design protection. The first reason may have some merit and will be discussed later in this chapter. The second reason has no merit, given the traditional ambit of industrial design protection. As pointed out at the start of Chapter II, industrial design protection has been provided to cover designs of primarily ornamental or aesthetic nature of which more than 50 copies are produced for commer-

---

10. These numbers were mentioned by Miller, 1974, p. 58.

11. This is not to say that industrial design protection has not been suggested. Indeed, it has been. See, for example, Melville, 1980, who favours industrial design protection for software for two reasons: (1) software is useful as opposed to fine (useless?) art; and (2) industrial design protection has a short term. As the discussion throughout Part I of this volume has indicated, neither argument is convincing.

cial purposes. Software and firmware are, by no conceivable stretch of the imagination, primarily ornamental or aesthetic. Incorporating them into the Industrial Design Act would, as with patent protection, drastically alter the current interpretations of and rationales for the Act.

Copyright Protection

The Copyright Act does not generally protect the ideas expressed in a creative work; it protects only the unique expression of those ideas. For computer software, this distinction between ideas expressed and the expression of ideas means that the algorithms themselves would not be protected by copyright. All that would be protected is a particular expression of an algorithm. Imitators would be free to use an algorithm so long as they developed their own expression of it and did not slavishly copy its expression from its originator. Copyright protection of software would be a very limited form of protection. To the extent that the commercial value of software lies in the ideas it expresses rather than its expression of ideas, copyright would offer no private or social benefits and trade secrecy would remain the preferred mode of protection. This statement is no more than a reiteration of the position of the authors referenced earlier in this chapter that the benefits of copyright protection would be small. It must also be remembered, however, that if the benefits would be small, the possible deadweight losses to society due to copyright protection would also be small.

Unlike patent or industrial design protection, copyright protection of computer software would not require that the interpretations of and rationales for the Act be substantially altered. Basically all that would be required would be an extension of the definitions of fixation and infringing copies to include new media for the expression of ideas. These definitions need only state that fixation or unauthorized copying would take place even if the media involved included new technology such as magnetic tape cartridges or read-only memories in silicon chip microprocessing units, regardless of whether the fixation or copy ever was in a form intelligible to the unaided human eye or ear and regardless of whether it had the potential of generating fleeting images on a video display unit or a printed copy. This extension of these definitions would make it clear to the courts that Parliament recognized the importance of new media for the transmission of expressions of ideas and would require similar recognition by the courts. Furthermore, it would be consistent with the extension of copyright protection to sound recordings, which represented a new medium for the expression of musical works.

The benefits of copyright. The method for estimating the social benefits was set out earlier in this chapter. It was shown that these benefits can be approximated by measuring the area of the shaded trapezoid in Figure 4. This area was shown to equal:

$$B = \frac{\Delta Q P^*}{E} + \frac{(\Delta Q)^2 P^*}{2Q^* E} \qquad \text{(Equation 6)}$$

It was also pointed out that if the values for ΔQ and P* are understated and those for E and Q* are overstated, the estimate of these benefits will be biased downward. This procedure is followed here because the estimates for the variables are imprecise, sometimes drawn more from anecdotal references than from published data because the data are not available. The purpose of this exercise is to show that even though the results will be biased against finding any significant benefits from extending copyright protection to computer software, these benefits will still be found to be positive. Thus, in a sense, a lower bound for these benefits is being determined.

In order to calculate B in equation (6), values for P*, Q*, E and ΔQ are needed. Each of these variables is discussed in turn.

(a) Average price of computer software packages. International Computer Programs, Inc., conducted a survey in 1978, the results of which indicated that about 41 per cent of the software products sold for less than $10,000, 66 per cent for less than $25,000 and 85 per cent for less than $50,000 (Frank, 1979). Interpolation between the numbers reported yields a median price near $15,000 in 1978. The distribution of price is undoubtedly skewed, more heavily represented by lower-priced software, and the median is consequently also below the mean. The use in this paper of $15,000 for P* in 1978 is, as a result, undoubtedly too low, biasing the results against finding sizeable social benefits.

In carrying out the calculations, an effort will be made to use Statistics Canada data reported for the six years 1972, 1973 and 1975 to 1978. Certainly P* increased during that time period as a result of inflation. Without any strong justification other than a desire to allow for inflationary forces, P* is made to equal $10,000 in 1972 and increases by $1,000 for each of the years for which other data are available. Overall, for the entire time period, P* equals $12,000. Despite the lack of hard evidence for this figure, it is plausibly low in light of the data available and should be adequate for calculating a downward-biased value for B.

(b) Number of transactions in computer software packages. Statistics Canada has reported the total sales revenue from software packages for the years 1972, 1973 and 1975 to 1978 (Appendix A). Dividing the revenues for each of these years by the average price of software for the respective year will yield a value for Q* in each year. The total for all of these years comes to 6,770. This figure represents the number of transactions, not the number of different programs. It includes each sale, even if one package was sold ten times, since the framework for analysis was constructed on the basis of Miller's results that it is particularly among multiple sales of software packages that costs of development and marketing can be reduced with the provision of copyright protection. Note, too, that to the extent that the estimate of P* is biased downward, the estimate of Q* will be biased upward, imparting a further downward bias to the estimate of B.

(c) Elasticity of demand. There are no estimates available of the price elasticity of demand for computer software. There are, however, numerous estimates of price elasticites of demand available for other products, a sample of which is presented in Table 21. The highest elasticity appearing in the table is 4.6, the short-run price elasticity of demand for fresh tomatoes, despite the availability of a wide range of substitutes in the form of other fresh, canned or frozen vegetables, not to mention other non-food uses of consumers' dollars (all of which would tend to push the elasticity upward). Another high elasticity is the long-run elasticity for motion pictures, which is only 3.7, again despite the wide choice of entertainment alternatives competing for consumer spending. It is difficult to believe that any closer substitutes exist for computer software (in the aggregate) than for motion pictures or fresh tomatoes. Given these figures, it might be reasonable to assume that the price elasticity of demand for computer software is no higher than 5.0, but the initial calculations will use E = 10.0, thus biasing the estimate of B downward even further.

(d) Increased number of transactions. At the start of this chapter, Richard Miller's studies were quoted extensively. They indicated that the percentage of firms which knew of no software that had not been developed and marketed as a result of the lack of protection had fallen from 87 per cent in 1973 to 74 per cent in 1977. It should be pointed out again that it is clear from the context of Miller's cover letter and his surveys that most respondents would interpret the phrase "legal protection" in his surveys to mean copyright protection. Based on these findings, the higher figure is used here (to bias downward ΔQ and hence B) and it is assumed that 13 per cent of the firms knew of at least one instance in which the lack of copyright protection proved to be a barrier to the production of a software package. Transferring this percentage to Canada yields an estimate that approximately 90 firms knew of instances in which software was not developed and marketed due to the lack of copyright protection. This paper does not, however, use 90 as a value for ΔQ. On the one hand, it is unreasonable to expect that in each of these instances the program would have reached the market even if copyright protection had been available. On the other hand, some of the firms presumably knew of more than one instance. And, finally, ΔQ should be a measure of foregone transactions, not foregone new programs as a result of the lack of copyright protection. Foregone transactions have two sources. The first is the foregone transactions from programs not developed -- presumably those which posed the greatest protection difficulties, in part, because of their broad applicability. It is reasonable to expect that each of these programs would have been sold a number of different times to different clients. For standard proprietary software packages, producers now estimate that they must have a large number of sales before a new pack-

Table 21

Selected Estimates of Elasticities

Category	Estimated elasticities	
	short run	long run
Food		
potatoes	0.3	
peas, fresh	2.8	
peas, canned	1.6	
tomatoes, fresh	4.6	
tomatoes, canned	2.5	
Nondurable goods		
shoes	0.9	
newspapers and magazines	0.4	
tires and related items	0.8	1.2
Services		
auto repair and related services	1.4	
radio and television repair	0.5	3.8
Travel and entertainment		
legitimate theatre and opera	0.2	0.31
motion pictures	0.87	3.7
foreign travel by United States residents	0.1	1.8
Public transportation		
taxicabs	0.6	
local public transportation	0.6	1.2
intercity bus	0.2	2.2
Utility services		
electricity	0.1	1.8
telephone	0.25	
Miscellaneous		
jewelry and watches	0.4	0.6

SOURCE: Miller, 1978, p. 120.

age is expected to be profitable.[12] The second source of foregone transactions is transactions that would have taken place had costs and prices of software been lower. To get an idea of the number of foregone transactions, it is assumed that only 50 new programs would have been produced and marketed had copyright protection been available. It is further assumed that each of these would on average have been sold four times, yielding a value for ΔQ of 200. This figure is probably an understatement of the true ΔQ in that it is calculated on the basis of small estimates of foregone transactions involving new programs and it completely disregards foregone transactions of existing programs.

Letting $Q^* = 6,770$, $P^* = \$12,000$, $\Delta Q = 200$ and $E = 10$ means that, from the formula for elasticity, ΔP equals about \$35. In other words, the values for these variables imply that, if copyright protection had been available, the cost saving and hence the price reduction would have been on average only \$35, a very small figure for programs costing on average \$12,000. This cost reduction seems particularly small in light of earlier mentioned anecdotal evidence that trade secrecy can sometimes add several thousand dollars to the cost of a program. In terms of Figure 4, ΔP is the vertical height of the trapezoid, B, which is to be measured. Finding such a small value for ΔP is further evidence of success in biasing downward the calculation of B, the social benefits of copyright.

The estimates of the values for each of the variables in equation (6) are necessarily crude because not many data are available for the computer software industry. To compensate for this lack of precision, defensibly plausible figures have been selected for each of the variables in such a way as to bias downward the estimate of the social benefits of extending copyright protection to computer software. Using these figures, the lost social value from not having had this protection is calculated as, at a minimum, slightly more than \$240,000 for the six-year period or roughly \$40,000 per year. If, instead, it is assumed that $E = 5$ or that $\Delta Q = 400$, these estimates would double to over \$80,000 per year. Although either of these changes might be a more reasonable reflection of the correct value, the figures less favourable to finding a high value for B have been used.

Suppose instead that one wished to establish that the expected benefits of extending copyright protection to computer software would be small (this, in fact, was the initial position of the authors). Using the same data sources, one would then attempt to select values for each of the variables which would bias upward the value of B and show that it was still small (i.e., one would attempt to show that the upper bound is low). In that regard, $P^* = \$20,000$ and $\Delta Q = 1,000$ are not unreasonably high; nor are $Q^* = 5,000$ and $E = 2$ unreasonably low. These figures generate, however, a value for B of \$11 million over 6 years or

12. See Frank, 1979, and Business Week, September 1, 1980, p. 54.

nearly $2 million per year! This figure is as much of an overestimate as $40,000 per year is an underestimate of the social benefits. The probability distribution of all the possibilities within this range is not known and it is therefore difficult to settle on any single number with confidence. It can, however, be asserted with confidence that the benefits are very likely to be greater than $40,000 per year (subject to later qualifications) and less than $2 million per year.

Administrative costs of copyright. In the fiscal year 1979-80, the Canadian Copyright Office registered 7,521 works and had a total budget of $336,000, yielding an average of only $45 per registered work.[13] Recalling that Canadian law does not require registration as a precondition for copyright protection, it can readily be seen that the average government cost per work protected by copyright is far less than even $45, as compared with over $350 per patent application. A further consideration is that the previous section was studying the number of transactions and not the number of new programs. If each new program were sold only four times, on average, then the average government cost for each new transaction would be certainly less than $12 per transaction.

Because registration is not likely to be required in Canadian copyright law, it is inconceivable that copyright protection of software would require more than $20,000 of government expenditure. This additional amount might possibly arise if computer software developers sought registration (to establish the presumption of originality) in such magnitudes that the Copyright Office found it necessary to hire an additional clerk and to provide additional filing capabilities. At an average registration cost of $50 per program, it would take registration of 400 programs per year to generate additional costs of $20,000 per year. Even in the United States, which generates substantially more software than Canada and which requires registration for copyright protection, the most programs registered for copyright protection in any year was 282, in 1972-73. Based on these figures, $20,000 per year is probably an overestimate of what the government's cost would be in allowing the copyrightability of software.

There are other potential social costs of copyright in addition to the government's administrative costs. The time costs for copyright protection, however, are negligible; the protection begins from the moment of fixation. Compared with patent protection, this feature of instantaneous protection favours both copyright and trade secret protection. Compliance cost comparisons are even more favourable to copyright. Roberts has estimated that application fees and costs for patent protection amount to more than $1,000 on average and that these costs are negligible for copyright protection (see Table 17).

---

13. Data provided by the Office of the Registrar of Copyright.

Enforcement costs of copyright. One of the questions often raised concerning the efficacy of copyright protection of computer software is its enforceability. It would be difficult to identify infringers and to prove infringement; programs could be altered sufficiently that they would not be construed as infringing copies and yet the commercial value of the program would have been misappropriated by the altered version. However, the jurisprudential history of copyright has established reasonably clear rules about what constitutes infringement. Translating a program from one computer language to another or transcribing the program from one medium to another would certainly constitute infringement. In the latter case, the analogy to music, perforated piano rolls and sound recordings is clear. Just as transcription from sheet music to piano rolls would constitute infringement,[14] so should transcription of software from printed form to magnetic tape or read-only memory on a silicon chip. To avoid possible misinterpretation, however, it was recommended earlier in this paper that the definition of infringing copies be extended to cover new media explicitly.

One of the techniques currently in use to facilitate enforcement of trade secrecy in software and of copyright in directories is the insertion of meaningless statements or data in the original.[15] In a computer program, statements can be inserted which the computer will never get to and never execute. In directories, fictitious names and addresses can be inserted. In both cases, infringing copies which have slavishly copied the original will contain the meaningless statements or fictitious entries. The infringer cannot then claim that his work was the result of independent effort on his part, for it is unlikely that in his independent effort he would have included precisely the same irrelevant material. That such techniques are available for both copyright and trade secrecy enforcement suggests two important and related points. First, both forms of protection pose enforcement difficulties. Second, independent effort can form part of a defence in each case.

Neither copyright nor trade secrecy grants a preemptive right of any sort. If someone can demonstrate that she wrote a computer program independently which happens to have numerous similarities with a program written by another person, she will not be guilty of either infringing a copyright or violating a trade secret. Under both types of protection, independent effort tends to create a presumption of innocence. Major differences between the two types of protection in terms of enforcement are that, under trade secrecy, access to the secret must be established and that, under copyright, damages are recoverable from infringing third parties. In practice, these differences are not great for, in the former situation, access is usually an important element of a

---

14. The White Music decision in the United States has effectively been reversed and changed through new legislation and judicial reinterpretation.

15. See, for example, the references cited in Part II.

copyright case and, in the latter, the law of torts can be applied to third parties which encourage the violation of trade secrecy.

The discussion of enforcement thus far suggests that copyright offers little advantage relative to trade secrecy. The differences are relatively minor, suggesting, as already indicated, that the expected net social benefits of copyright protection vis-à-vis trade secrecy would be small. These minor benefits would, however, be the result of the creation of additional and/or more valuable and/or more widely disseminated software. Regardless of the source of the social benefits, they would likely occur in conjunction with additional enforcement costs, if for no other reason than that additional programs would be available. The paucity of cases involving trade secrecy or copyright and computer software, especially in Canada, makes it difficult to estimate these additional enforcement costs. The fact that they do not exist does indicate, however, that the small increase in the number of programs would be likely to result in very few additional suits and to impose small additional social costs in the form of enforcement costs. To the extent that copyright enforcement might be cheaper than trade secrecy enforcement, the additional costs resulting from enforcement of more cases would be ameliorated by lower enforcement costs in each case.

Other problems with copyright

(a) Term of protection. It has been argued often in this study that software and especially firmware are analogous to music and sound recordings. Following that analogy, it seems reasonable that if copyright protection were extended to include computer software, the term of protection should be 50 years from the date of creation rather than the longer term of life of the author plus 50 years. Fifty years is a long time to grant intellectual property protection for computer software which, by even the longest estimates, has a durability of no more than 20 years. If, however, no program has a durability of greater than 20 years, then a 50-year term of protection is unnecessary but it is also irrelevant, having zero social costs and zero social benefits for the period beyond 20 years. Comparing the term of protection with the durability of software is, at any rate, an incorrect approach to setting the term of protection. The optimal term depends not on what the current state of the art is but on what it will be or could be in the future and on what incentives society wishes to create.

A term of protection of five years could be established and that would create an incentive to produce short-lived software. A longer term of protection would offer more incentive to work on software with greater durability. The shorter terms would also encourage firms to eschew copyright protection in favour of trade secrecy in the hopes of maintaining monopoly control of their property beyond the term of copyright protection, whereas a longer term of protection would encourage less use of trade secrecy and greater dissemination of copyrightable works. Traded off against these benefits of longer terms of protection are the social costs of granting extended monopoly protection unneces-

sarily to works which would have been created even if the term of protection had been shorter. Balancing these benefits and costs is extremely difficult. Articles on the optimal life of patents abound and their discussions can easily be applied to copyright protection as well.[16] The major difference is that the monopoly granted by copyright protection is extremely limited. Consequently, the social costs of a term of protection, which might be longer than necessary to elicit creative works, would be extremely small. As a result, there seems to be no reason not to afford software the 50-year term of protection, but it is also evident that a shorter term of protection would probably not have a much different effect on social benefits and costs. At a minimum, though, the term of protection should be at least 25 years for two reasons: (1) to encourage longer-range development and wider-range dissemination; and (2) to satisfy Canada's obligations under the Universal Copyright Convention.

(b) International obligations. Canada is a signatory to two major international copyright conventions: the Berne Convention and the Universal Copyright Convention. One of the basic provisions of these conventions is that foreigners be afforded the same treatment as nationals for copyright purposes. Three questions arise regarding Canada's participation in these agreements with respect to computer software:

1. If Canada were to extend copyright protection to computer software, would we also be required to provide protection for foreigners under the conventions?

2. If Canada is not required to provide protection for foreigners, should we limit protection to software developed only in Canada?

3. If we are required under the conventions to provide protection for foreigners, would a trade imbalance in software reduce the net social benefits to Canada sufficiently to justify not extending copyright protection to software for anyone?

It is reasonably clear that the international copyright conventions do not require the protection of computer software (Keyes and Brunet, 1977, p. 21). It is also reasonably clear, however, that any member country which does extend copyright protection to computer software must do so for the nationals of all member countries (Torno, 1978). Because the authors are reluctant to recommend that Canada breach its international treaty obligations, question two is left unanswered.

So far in this discussion, Canadian software producers have been treated as if they operate in isolation from the rest of the world.

---

16. See, for example, the extended discussion in Scherer, 1980.

Such is not the case. Some Canadian firms produce and market software for customers in many countries and some Canadian customers purchase software which has been developed and marketed primarily in other countries. Once these international trade flows in computer software are taken into account, the estimates generated so far will require qualification.

The first step in the analysis of international considerations is to set out more specifically the assumptions implicit in the calculations made earlier in this chapter; then the effects of altering these assumptions are explored.

In calculating the area of the trapezoid in Figure 4, it was assumed implicitly that all Canadian software is produced and marketed domestically, that no international trade takes place and, hence, that the size of foreign markets and their legal environments would be irrelevant. Furthermore, because some of the data used were taken from the results of U.S. surveys, there was also an implicit assumption that the U.S. industry makes its decisions without concern for the Canadian market or legal environment. Only to the extent that these assumptions are correct are the calculations reasonable.

It is interesting that the initial assumptions can be independent of the national origin of the firms in the industry. To see this point, suppose that all of the firms in the Canadian software industry were domiciled outside Canada, but that they marketed specific programs only in Canada. Canada would thus have no exports of computer software, only imports; nevertheless, the gains to Canada from copyright protection of computer software would be exactly the same as those in the previous example. Copyright would reduce, on average, the costs of producing and marketing programs in Canada and, due to the forces of competition, these cost savings would be passed on to Canadian customers. This example highlights the irrelevance of both the balance of trade in computer software and the domicile(s) of software producers when considering the costs and benefits of copyright protection. Instead, as will be demonstrated below, the important question involves the size of the Canadian market relative to other important markets and, to modify the assertion in the previous sentence slightly, the interaction between relative market sizes and the balance of trade.

Suppose now that software producers all operate in the world market for software. Suppose further that Canada accounts for only a negligible portion of the entire world demand for computer software. Enacting explicit copyright legislation (or any form of intellectual property protection) in Canada would thus have at best a negligible impact (and possibly a detrimental impact) on social welfare in Canada. To see this point, consider two additional possibilities, this time concerning legal environments in the rest of the world.

Suppose first that neither Canada nor the rest of the world has copyright protection for computer software. In this case, trade secrecy

would be practised everywhere. If Canada were to enact copyright protection, its doing so would have no impact on production and marketing decisions because firms would not be willing to give up trade secrecy and reduce their costs in a small market (Canada) when doing so would greatly increase the risk of misappropriation in the rest of the world. In this case, Canada would get no benefits from having copyright protection but would still have the administrative costs (albeit small since the system would be dormant).

Alternatively, suppose that all other countries offer copyright protection for computer software but that Canada does not. Under all of the suppositions stated above, the possibility that software can presently be expropriated by imitators for the Canadian market represents such a small portion of world sales that it would not seriously affect the profit-maximizing calculus of the software producing firms. The software producers might be thought to have a choice between: (1) using trade secrecy in Canada along with selling at a higher price in Canada to cover the higher costs of trade secrecy, and (2) eschewing trade secrecy in Canada but allowing misappropriation to occur. This is not, in fact, a choice at all. If copyright were available in the rest of the world and if Canada represented a negligible amount of software purchases, practising trade secrecy in Canada would serve no profitable purpose beyond that served in the rest of the world. In other words, to the extent that copyright allows software producers to reduce their costs of protecting their software (vis-à-vis trade secrecy) in the rest of the world, it also makes this software available for misappropriation in Canada regardless of whether trade secrecy is practised in Canada. In this situation, Canada can exercise its option of being a free-rider and can utilize computer software developed elsewhere at much lower cost than it would incur if it were to develop the software itself.[17] Software in Canada would be cheaper than software anywhere else in the world. Enacting copyright protection would raise the price and reduce the quantity of software used in Canada; that is, it would actually reduce social welfare.

The foregoing analysis shows that if Canada accounts for a negligible share of the world purchases of computer software and if there is free trade in a world market for software, Canada would certainly not benefit from extending copyright protection to computer software. Moreover, it might well be considerably worse off than it would be if it could expropriate software from other countries without fear of retaliation in other areas.

---

17. The force of this argument against extending copyright protection to computer software is considerably dampened when one asks further how much misappropriation would take place in Canada under these suppositions. The amount of misappropriation activity would probably not differ a great deal from a situation in which all countries had copyright protection, recalling that, in the preceding section (ignoring international trade), such relatively low estimates were obtained for the lower bound of the benefits of copyright protection.

To demonstrate the conditions under which copyright of software in Canada would not generate any benefits for Canadians, a number of highly restrictive conditions have been created. When these conditions are relaxed, it becomes clear that copyright could generate some benefits. What cannot be established, however, due to the lack of data, is the size of the benefits (or, more precisely, how much the earlier estimate of these benefits should be reduced). In the first place, there are some barriers to international trade in computer software by virtue of the tariffs and/or other import restrictions in many countries. Furthermore, communication and transportation costs contribute to limiting the market from a world scope to continental, if not national, boundaries. Finally, due to different legal, social and cultural environments, the demands for different types of computer software will vary across national boundaries, contributing even further to the creation of national as opposed to broader markets. The effect of these forces is to make the Canadian market more relevant for Canadian software producers than would otherwise be the case. To the extent that this occurs, benefits can be created for Canadians by enacting copyright protection for computer software. In the second place, it may be stretching reality to say that the Canadian market has a negligible impact on decisions made by software producers throughout the world. At the margin, being able to protect the intellectual content of computer software in Canada will affect some decisions about development, marketing, costs and prices and it will consequently generate some benefits for Canadians.

In general, then, this analysis of international considerations indicates that a minimum estimate of the area of the trapezoid in Figure 4 is probably less than $40,000 but how much less (and even whether it exists at all), is not known.

Consider, as an example, the possibility that only ten per cent of the benefits measured earlier would actually be created by Canadian copyright because Canada is such a small portion of the world market. In this case, the social benefits from copyright would be a minimum of $4,000 per year but the administrative costs could still be as high as $20,000 per year if there are indivisibilities in the administration of sections of the Act. Canada would lose $16,000 per year. Alternatively, using the upper bound estimate of the benefits, ten per cent of these is still $200,000 per year, generating substantial net gains for Canada.

These numbers pose an interesting problem: it is difficult to reject two competing hypotheses with any confidence. To establish that the net benefits would be positive, one must examine the lower bound of the estimate of the benefits and, on balance, this lower bound could be as low as -$20,000 or even considerably lower, indicating the difficulty of establishing the existence of these benefits. Yet to establish that the net benefits would be less than zero, one must examine the upper bound of the estimates and plausible estimates of it seem to be cer-

tainly greater than $100,000 and perhaps as much as $2 million per year. In neither case can the hypothesis be rejected. In the former case, it is difficult to reject the hypothesis that the net gains would be negative. In the latter, it is difficult to reject the hypothesis that the net gains would be positive.

(c) Damages. Sections 20 and 21 of the present Copyright Act provide that a successful plaintiff may recover: (1) lost profits, plus (2) whatever part of the defendant's profits the court may decide to be just and proper, plus (3) all infringing copies and plates or the conversion thereof (i.e., the revenues earned by the defendant's sale of infringing copies, plates, etc.). These damage awards provide, essentially, for potentially large punitive damages by allowing the plaintiff to recover possibly far more than just his or her lost profits. Punitive damages have the social benefit of discouraging illegal activities but the social cost of encouraging damage suits with a low probability of success. Without entering the debate on optimal enforcement and penalties, it should be noted that Keyes and Brunet seem favourably disposed to a clearer statement in the law of the allowance of punitive damages and their recommendations seem acceptable. The current Act does not allow damage awards which would be more than socially optimal. Furthermore, any criticism of potential copyright damage awards is not, per se, a criticism of extending copyright protection to computer software but is instead a criticism of the Copyright Act itself.

(d) Compulsory licensing. The rationale underlying compulsory licensing is that it can facilitate the flow of information by reducing the transactions and negotiations costs between originators and those who wish to use the protected material. The basic provisions of the Copyright Act allow for potential producers to apply to the Minister of Consumer and Corporate Affairs Canada for a licence if the copyright owner does not make a copyrighted book available at all or in "sufficient" quantities in Canada (sections 13 and 14). Only on the rarest of occasions would the copyright owner not reach an agreement with a marketing agent for a work which had market value but the provisions of sections 13 and 14 of the Act could be extended to software as well as to books.

Section 19 of the Act deals with a different form of compulsory licensing for sound recordings. It provides that so long as the copyright owner has reproduced a work anyone else may reproduce it upon payment of a royalty of two cents per playing surface or per mechanical contrivance. It is difficult to see that this scheme would increase the social benefits of software production development, despite the fact that sound recordings have been used here as an analogy to software and firmware. One possible concern is that a royalty payment of only two cents per contrivance (silicon chip?) would likely be such inadequate compensation that firms would not be compensated fully for the value of their intellectual property. Copyright would then not be able to create the requisite incentives for generating the social benefits described

earlier in this chapter.[18] There is no reason, though, to expect that royalty payments will be fixed at two cents ad infinitum. More importantly, compulsory licensing can be economically efficient only if the administrative costs of the scheme are less than the transactions costs of using the market. It is understandable that the transactions costs of collecting royalty payments for each use of a musical composition might be very high relative to the costs of administering a compulsory licensing scheme. But even for products such as photocopies, it is not clear that such is the case.[19] There is no reason why, in the market for computer software or even firmware, the transactions costs of collecting royalty payments through the marketplace should outweigh the costs of administering a necessarily less efficient scheme for compulsory licensing.

(e) Fair dealing provisions. Section 17(2)(a) of the Copyright Act states that "any fair dealing with any work for the purposes of private study, research, criticism, review, or newspaper summary" does not constitute an infringement of copyright. Fair dealing, like most intellectual property law, has been incorporated into the law to allow for widespread dissemination of ideas at low social cost. In this case, copying short sections of a protected work for research or review purposes will have the social benefit of encouraging research and writing and generally the dissemination of ideas. At the same time, it is unlikely to have a detrimental effect on the creator's revenues. In the absence of the ability of the creator to practise price discrimination, fair dealing provisions are usually justified when both the social costs and benefits are considered (Liebowitz, 1980a).

Allowing fair dealing provisions for computer software seems questionable. On the one hand, there should be no objection to allowing researchers to copy portions of the software for private (and personal) study and review once the software has been developed and marketed. On the other hand, the loss of a single sale of the software could result in the loss of revenue to the developer of thousands of dollars. If fair dealing provisions are allowed for computer software, they should be limited specifically to personal study and research concerning the software itself, and they should not include study and research which uses the software for the study and research of other questions.

---

18. See Keon, 1981 and Blomqvist and Lim, 1981.

19. See the first section of Chapter II. These ideas have been more carefully developed in Liebowitz, 1980(a).

(f) The problem of firmware. The most recent legal case involving computer programming and copyright is the Data Cash case.[20] This case highlights the problem of intellectual property protection for a product which is mass marketed and which consequently is not economically amenable to trade secret protection.[21] A computer program which is produced on a silicon chip can readily be copied, as was demonstrated in that case. The chip, which embodies the results of intellectual and engineering effort, can easily be appropriated by persons or firms who did not invest in the development of the program embodied in it. The appropriating firm can become a free-rider on the original investment in the program and even, in some measure, on the original marketing effort. The firm developing the program, programming it onto the silicon chip and then marketing the chip or the product using the chip has made an initial investment, little of which must be incurred by the appropriator. The latter firm, in the simplest case, need only purchase the product, remove the chip, contract for production of identically programmed chips, change the external design of the product and market the product itself.[22] And, in situations involving hardware with interchangeable plug-in chips or tape cartridges, the appropriating firm need not even design a new product; it need only market its own copies of other chips or cartridges already on the market.

---

20. Data Cash Systems, Inc. v. JS & A Group, Inc. (N.D. Ill., Sept. 26, 1979 , No. 79-591). The details of this case, along with a legal discussion of the judgement, were presented in Chapter II.

21. It will be recalled from Chapter II that trade secret protection is a close substitute for copyright protection for products which involve explicit contractual arrangements between the vendor and the user. For products which are mass marketed without contracts, however, trade secret protection is not a viable means of protecting one's intellectual property. In the Data Cash case, it would not have been reasonable to expect Data Cash to require all purchasers of their computer chess game to sign non-disclosure agreements because if one of the purchasers of the game did disclose it (i.e., use it to copy from) the problems of enforcement could become virtually insurmountable. If, for example, a customer purchased the game in good faith and made a present of it to another person, would that act in itself constitute a breach of the non-disclosure contract? Probably not. If the recipient, not a party to the agreement, then used it to copy from, there is no legal breach of contract. Yet such a transaction and the possibility of its recurrence greatly diminish the value of the intellectual property embodied in the game.

22. These steps were basically those followed by JS & A Group, Inc. in the Data Cash case. The defendants demonstrated remarkable alacrity in marketing their chess game only a year or so after the Data Cash game was introduced. The only reason to change the external design is to avoid prosecution under the Industrial Design Act or the Trade Marks Act.

In many respects, firmware and software are very similar. They basically use different media for the production and storage of computer programs. In economic terms, for the purposes of this discussion, it is the potential for mass marketing firmware which differentiates it from software, and then only in degree. Because of this potential for mass marketing, firmware cannot easily be protected using trade secrecy, whereas software marketed on a very small, limited scale can make use of trade secrecy with less costs. In terms of Figure 4, the costs of protecting firmware with trade secrecy are much higher than the costs of protecting most traditional software and so offering it copyright protection will provide a larger cost saving and a larger trapezoid of social benefits. None of these benefits was included in the measurement of the social benefits.

(i) Patentability of firmware? Because new firmware is a new physical product, embodying an idea, one might expect that firmware would be patentable. Under current patent law, though, it would be difficult to make a case for patentability of firmware. For a product to be afforded patent protection, one essential criterion is unobviousness and it is not clear that any particular chip programmed in one specific way is unobvious. Generally, chips are nothing more than micro hardware and can be programmed in many ways. The three analogies that follow emphasize the lack of unobviousness of firmware.

The first analogy concerns an architect and a builder. An architect can have an intellectual property right via copyright in his or her plans for a building. Neither the architect or the builder can, however, patent the building constructed on the basis of the plans.[23] The reason is that the building itself is obvious; a competent builder could use the plans to construct the building. Similarly, a competent electronics firm which is normally in the business of manufacturing firmware could produce firmware embodying a new program. The firmware itself does not satisfy the criterion of unobviousness even though the result of the intellectual effort underlying the program on the chip or tape might be considered novel.

A second analogy comes from the rather obvious example of book publishing. A book is not patentable because it is no longer unobvious. The expression of ideas within the book may well be novel and unobvious but to reproduce this expression in book form is by no means unobvious and hence is unpatentable. It can be argued quite persuasively that programmed silicon chips and tape cartridges are similarly not a subject for patent protection because of their obviousness, despite the possibility that they may embody novel and unique expressions of ideas (i.e., unique computer programs).

---

23. See the previous sections on copyright and patent. Note the differences between Canadian and U.S. law with respect to copyright protection of architectural plans.

The third analogy involves sound recordings, which are probably closest in form to computer firmware. Traditional forms of sound recordings are no longer patentable because they are neither novel nor unobvious as an art form or as a medium of expression. The composer has been given an intellectual property right through copyright, though. What is protected by copyright is the expression, not the specific product. The product is not patentable because a competent sound recording manufacturer could, via prior art, manufacture the sound recording. The recording consequently is not unobvious. Nevertheless, the particular composition is a unique expression and hence qualifies for copyright protection. And the amount of monopoly power created by the grant of copyright protection is not great since others are free to emulate the style of music on different sound recordings, creating a large number of close substitutes, all of which are copyrightable.

The point of these analogies is quite simple. Computer firmware, such as silicon chips and tape cartridges, does not pass the test of unobviousness. Consequently, computer firmware is unpatentable and should remain so.

(ii) Copyrightability of firmware? It is time for society to recognize that the media for the expression of ideas are continuously expanding as a result of technological change. The law has had to make relatively substantial adjustments to technological progress in the past and will undoubtedly be forced to do so in the future. Composers were once able to receive rewards for their artistic efforts simply by charging a price for printed copies of their music. As the era of sound recordings developed, composers' rewards were seriously jeopardized. Without the extension of intellectual property rights to cover sound recordings, "bootleg" recordings could be mass-produced without payment to the original composers. Yet copyright law had until then addressed only visible copies, not audible copies. A reinterpretation of the scope of copyright was necessitated by technological change to include audible as well as visible copies.[24]

The scope for copyright protection will continue to need reinterpretation as new technologies are discovered and become economically

---

24. In their earliest stages, sound recordings were patentable. Only the technology of the industry was patentable, however -- not each different recorded performance (disc vs. cylinder recordings, or wax-process discs vs. direct-cut discs). See Gelatt, 1966; Read and Welch, 1959; and Batten, 1956. The analogy between the developments in the sound recording industry and the firmware industry should be clear. Silicon chips or cartridge tapes, per se, are patentable, as are the different processes of producing them, just as different types of sound recording media and their methods of production were patentable. But what went onto the sound recordings has never been patentable; nor should the programs be when they are embodied in firmware.

feasible. Even if a particular medium never produces a copy of an expression of an idea which is comprehensible by the unaided human eye or ear, it must still be recognized as a medium for the expression of the idea. Without contesting the jurisprudential declaration that copyright protection should not be afforded to ideas, per se, it can be argued that copyright protection should be made available for particular expressions of an idea, including expressions in the form of computer firmware such as tapes and chips. Tapes, chips and other possible future embodiments of software are analogous to books and sound recordings. All of these products, including firmware, are essentially media for the unique expression of an idea and deserve equal protection under the law. In each case, if product A is deemed to be a copy of product B, the producer of A is or should be liable to copyright infringement suit.

(iii) The determination of infringement. The question of what constitutes infringement is often a thorny one. Guidance as to what would constitute copyright infringement for firmware should be taken from the case law concerning sound recordings. Certainly, copying the program from one chip onto a different coloured or shaped chip would constitute infringement. Also, a demonstration that the programs embodied in the firmware were virtually identical should constitute a rebuttable presumption of infringement.[25] A defence would be to show independent effort to design the chip. It is conceivable that two firms working independently could design programs for chips utilizing similar algorithms and copyright protection would not give preemptive rights to the algorithm to one of the firms. It would give them protection only for theirunique expression of the algorithm, as is presently the law for most copyrightable materials.

The recognition that an algorithm is neither patentable nor copyrightable highlights the fact that copyright protection for computer firmware would be of very limited benefit to firmware developers. It would protect only their unique expression of an algorithm. Other firms could examine the firmware, detect the algorithm and design their own unique expressions of the algorithm. The original effort of divining the algorithm itself is not protected, nor is the generalized value of the marketing effort. A copying firm may have to replicate some of the original effort to discern the algorithm(s) embodied in firmware, but it would not be likely to undertake this effort if it expected that independent discovery of the algorithm would be cheaper. Consequently, copiers of the algorithm may be at some advantage over original firms in the realm of development costs. Copiers may also have some advantage to the extent that they are able to ride free on some of the marketing effort of the original firm. They would also perhaps have an advantage if they could, by being second in the field, produce firmware which in some way improved upon that of the original firm.

---

25. Computer specialists and electrical engineers have emphasized to the authors the relative ease with which software can be copied from firmware.

The point is that copyright of firmware would create an extremely limited monopoly. The deadweight losses to society resulting from the monopoly would be quite small due to the potential for competition. Although a cynical reader might assert that the benefits of firmware copyright protection would be zero, it is this paper's contention, based on an argument from analogy and an argument from economic theory, that the benefits would be positive.

(iv) Benefits of firmware copyright. The first author to write and publish a successful book on, say, bicycle repair would receive a reward commensurate with the quality of and demand for his product.[26] This reward will undoubtedly be eroded over time as other authors compete away his rents. The topic of the later books would be the same -- bicycle repair. The table of contents may have the same structure and even the steps in the repair procedures may be identical. That is, the algorithms may be identical in all books published on the topic of bicycle repair. Nevertheless, each author has an intellectual property right in his or her particular expression of the algorithm. Each receives a benefit to the extent that no one else may copy his or her unique expression of that algorithm. It is not to be denied that the later authors benefit from the initial dissemination of the algorithm and from the demonstration of marketability by the original author and are free-riding to some extent. Nevertheless, each author must put forth an independent effort to provide a unique expression of the algorithm and receives protection for that unique expression. This protection may be of small benefit, but it most certainly is positive.

Computer firmware is analogous to such books. The reward for discovering the algorithm comes from being first in the field and being able to exploit a short-term monopoly until other firms are able to design and produce similar firmware. Even after other firms have entered the market, though, the original firm would still have a property right in its firmware, with its own uniqueness and its own marketability.

The theoretical argument that the benefits are positive follows from the above analogy. It is nothing more than a recitation of the case for copyright of books. Bestowing a property right on a particular expression of an idea confers benefits on the originator of that expression. It imposes costs on potential appropriators of the idea in that they must devise their own expressions of the idea. Hence, it raises the costs and prices of free-riding substitutes for the original expression of the idea. Consequently, it is clear that the private benefits are positive when compared with a situation without intellectual property protection. With regard to social benefits, it was argued in Chapter II and earlier in this chapter that when the private benefits of copyright protection are positive, generally the net social benefits are too.

---

26. This example was suggested to the authors by Stan Liebowitz.

(v) Interchangeable vs. hardwired firmware. Firmware falls into two broad categories: that which is wired directly into other hardware components (i.e., hardwired) and that which is interchangeable between units (e.g., plug-compatible microprocessing units or tape cartridges). It has already been argued that, regardless of whether the firmware is interchangeable or hardwired, it is still an expression of a computer program and, as such, should not be accorded patent protection. The analogy between firmware and sound recordings is consistent with this recommendation. Interchangeable firmware is much like sound recordings in that differently programmed firmware can be used with the same hardware just as different sound recordings can be used with the same playback system. Hardwired firmware is, however, more analogous to a record player with the record built in and able to play only one record -- more like a music box, really.

Copyright protection should not be extended to machines or to devices. These are and should remain eligible for consideration for patent protection. But if an individual produces and markets a music box which plays one particular song (which no one else has put onto a music box) she cannot get patent protection for this "novel" music box, even if the song is an original composition. Music box technology has existed for approximately 500 years! If someone were to transcribe the song from the music box to a symphonic orchestration, however, she probably would have copyright protection for her song. And certainly if she had built the music box to play a previously copyrighted song, she would be liable for copyright infringement. Firmware is very close to sound recordings in many of its characteristics and would not necessitate much revision or reinterpretation of the law to be copyrightable.

(vi) Medium of expression vs. new product. As has been said previously in this paper, copyright protection should not be extended to new machines or devices. It has consistently been argued that new firmware is a new expression of an idea using prior art but this argument should not be generalized to include new products and devices which should more reasonably continue to be protected via trade secrecy. For example, the notion that for intellectual property purposes soft drinks are a medium for expressing a secret formula for Coca-Cola is unacceptable, as is the notion that fried chicken is a medium for the expression of the secret recipe for Kentucky Fried Chicken. These are but two of many possible extensions of the position taken in this paper that would be inappropriate. Sound recordings are protected by copyright internationally because a market failure became apparent. Similarly, a potentially serious market failure in software, and especially firmware, has become apparent and this has led to the recommendation that software also be granted legislative intellectual property protection. No such market failure (in terms of creative incentive) has been demonstrated in the soft drink or fried chicken "media" and, until a market failure is apparent for any product and its related media, no legislative intellectual property protection should even be considered.

(g) Copyright and industry structure. In their discussion of copyright and the structure of the computer software industry, Braunstein et al. have written:

> [T]he use of trade secrets can lead to excessive concentration in the industry. The argument can be summarized as follows: Lack of information about existing packages places a premium on the control of a large portfolio of those packages whenever there are some complementarities between various items of software. In the literature, those gains are referred to as economies of scope. Economies of scope exist whenever the total costs of producing two or more items of software in separation exceed the total costs of producing them in one firm.
>
> The complementarities between various items of software may be attributable, for example, to the interchangeability between subsets of computer programs. If a firm cannot obtain the information about other programs which can be useful in the development of the new one then it will have to waste resources on reinventing the existing complementary software. A firm with a large portfolio of programs is at a comparative advantage over a small firm with the proprietary programs. If copyright protection is substituted for trade secrecy then the flow of information is facilitated and firms can fully utilize the existing stock of knowledge.
>
> Disclosure also facilitates competition because the downstream firms -- the purchasers of software -- have better knowledge of the available products. An improvement in consumer information should drive down the price. More important, it may also ease entry into the upstream -- software -- industry. Then the argument is that with better information the downstream firms can more easily search out alternative sources of supply. Hence, the well-established firms need not have a special advantage over the newcomers because the newcomers can advertise widely and specifically their products without the fear of losing trade secrecy protection. This discussion brings out a point which curiously seems to have been missed in the current literature, namely that disclosure not only reduces wasteful duplication of the research effort, but also reduces search costs for the ultimate buyers. This reduction in search costs facilitates improved matching between buyers and sellers and improves social welfare. (Braunstein et al., 1977, p. III-13, 14)

This analysis is consistent with the position taken in this paper. One possible concern in the minds of some readers might be, however, that if the trends in the software industry continue such that marketing of full-lines will be important, then despite the arguments of Braunstein et al. it is possible that copyright protection will favour those larger firms producing proprietary packages of general applicability. In effect, the argument might go, copyright protection could reduce the size of one type of barrier to entry and growth -- namely, a barrier existing due to fear of misappropriation -- and replace it with another -- a financial barrier due to the required large-scale marketing for proprietary packages. Furthermore, there is a reasonable concern that, due to indivisibilities in the legal process, larger firms may have a cost advantage over smaller firms in copyright infringement suits or even in the use of threats of infringement suits as a harassment tactic. Additional concern arises because in the United States 971 out of 1,205 programs registered with the Copyright Office were registered by IBM and Burroughs (CONTU, 1977, p.85).

It should not be surprising that hardware manufacturers (and dominance in this industry is on the wane) should be most likely to pursue and receive the benefits of software copyright protection. These firms are attempting to provide general-purpose, wide-use software for many different potential hardware customers. They are the software developers most likely to be developing and marketing software which can be most easily used by a large number of customers (i.e., which can most easily be misappropriated without competition and, hence, which would benefit most from copyright protection). Their size relates to the possible existence of economies of scale in mainframe production. This size makes it more efficient for society if they are the firms which develop and market the software most susceptible to uncompensated expropriation. As Demsetz has demonstrated in his discussion of the concentration-profits literature, concentration and bigness often are the result of efficiency and should be less of a concern to society (for efficiency reasons) than they typically have been (1974, pp. 164-183). Demsetz's arguments are correct; therefore the size of the beneficiaries of copyright protection should not enter into a decision of whether to implement copyright. As has been stated, the net social benefits may well be small and they certainly would not all accrue to the large mainframe producers. Even these producers compete with other software developers and with each other. Hence, a substantial portion of the benefits of cost reduction would be passed on to software users due to the forces of competition. Furthermore, to the extent that copyright would foster the generation of new software, computer software users (and their customers, throughout the general equilibrium system) would benefit from having additional, cost-reducing software available.

For firmware, the story may be somewhat different. Firmware markets will depend increasingly on sophisticated marketing techniques. Acquiring these techniques and financing them may pose more of a barrier to entry and growth by small firms than would be the case with traditional software. Chapter I indicated that the technology of pro-

gramming firmware and producing it would not impose serious barriers to entry or create increased concentration. Only if financial capital markets have serious imperfections would the more sophisticated marketing techniques, which might be necessary for some firmware, create additional barriers to entry or growth and lead toward increased concentration in the industry.[27] Certainly, in the absence of copyright protection of firmware, larger firms with well-established marketing techniques and networks would be in a stronger position to expropriate software and market it as their own. Consequently, (aside from the possible use of legal harassment by larger firms) extending copyright protection to firmware would probably protect less experienced firms while they developed their markets and would lead to a less concentrated firmware industry.

New Legislation

The potential problems with extending copyright protection to computer software have led many people to suggest that new legislation be drafted to provide intellectual property protection of software. One of the leading proponents of new legislation is the World Intellectual Property Organization (WIPO), which has stated:

> Whereas patent law protects the technical idea underlying an invention, copyright law focusses on protecting the form in which ideas are expressed, although protection is not limited to that form. Thus, copyright protection would seem to be particularly appropriate for computer software as a whole (and not merely computer programs) since a large amount of computer software consists of descriptive or explanatory matter; even a computer program (consisting for example, of magnetic tape) is a form of expression -- of the ideas contained in the software leading up to the program. In most cases the intellectual creativity in computer software resides in the skill and effort used to make those ideas "understandable" to a computer, as economically and as effectively as possible. However, although some kinds of computer software (especially those in verbal form) are clearly protectable under copyright laws, experts disagree on whether other kinds (particularly a computer program, on magnetic tape for example) can be considered a literary, artistic or scientific work, which are the traditional subjects of copyright protection. Moreover, such protection may be of very limited value since it essentially covers only copying (or related acts such as translation or adaptation); thus, in itself, the use of

---

27. Yale Brozen has argued that advertising and other marketing techniques more often than not provide a vehicle for entry as opposed to a barrier to entry (1974, pp. 115-136).

> a program to operate a computer cannot be prevented by copyright law (just as the making of a cake cannot be an infringement of the copyright in the recipe). It is essential that use in a computer should be covered by the rights in computer software; it is, in fact, possible that copyright law can provide a remedy in this case since it is probable that the use of a program always involves its copying in the computer memory, but the courts may not regard such internal reproduction as sufficient for the purposes of copyright law. The model provisions essentially adopt a copyright law approach which takes account of their subject matter's affinity with copyright protection and overcomes the possible limitations indicated above. (WIPO, 1978, p. 5)

The WIPO discussion of copyright is not very clear on just what it perceives the "possible limitations" of copyright to be. The discussion in the previous section of this chapter probably covers most of the problems they anticipated. These problems have been shown not to be serious if, indeed, they are problems at all. An examination of the Model Provisions on the Protection of Computer Software written by WIPO (see Appendix B) reveals some provisions which could be incorporated into copyright legislation and some which are unnecessary for Canada.

It has already been indicated that the WIPO definition of software is less than completely satisfactory because of its lack of clarity with regard to firmware. The "Rights of the Proprietor" in section 5 of the Model Provisions, covering unauthorized use and disclosure are, in essence, an attempt to create federal legislation for trade secret protection of computer software. Chapter II of this paper showed that provincial trade secret protection is adequate in this regard and that there is no conflict between provincial trade secret protection and federal copyright protection. Consequently, these sections of the Model Provisions are unnecessary in Canada.

The only other major difference between the Model Provisions and the Canadian Copyright Act is in the term of protection. WIPO recognizes that computer software durability is likely to increase in the future, yet it is reluctant to grant long-term protection to computer software. As has been said here, the potential social costs of allowing a 50-year term of protection have been grossly exaggerated. But if Canadians are not persuaded by this argument, there should be no difficulty with creating computer software as a separate class of copyrightable material, as has been done regarding sound recordings, and granting it a shorter term of protection. There is some indication in the Universal Copyright Convention, however, that a minimum term of 25 years would be required by Canada's treaty obligations.

In conclusion, although there can be no strong objection to creating new legislation along the lines of the WIPO Model Provisions on

the Protection of Computer Software, there is no compelling reason to do so. The protection sought by WIPO is, by their own admission, a form of copyright protection. Computer software and firmware are appropriate subject matter for copyright protection. There is little difficulty with incorporating computer software into the Copyright Act.

Chapter IV

SUMMARY AND CONCLUSIONS

The invention and growing popularity of sound recordings markedly altered the ways in which composers were and were not compensated for their creative efforts. Nevertheless, a study of the sound recording industry in 1915 would have found it to be growing and developing rapidly. Arguments against copyright in the industry would have pointed to this growth, questioning the existence of a market failure. But a market failure did exist even though the industry was so vibrant. Copyright protection was extended to cover this new medium for the expressions of ideas to correct the market failure.

It is time for Parliament to consider seriously once again the extension of copyright protection to cover new media for the expressions of ideas to correct new market failures. In doing so, it must be careful to include protection for all software, including firmware, and it must also take care to extend only those provisions of the Copyright Act to software for which the net social benefits would be positive.

Using a basic supply and demand approach, the minimum social benefits of providing copyright protection for traditional software were calculated to be approximately $40,000 per year. Substracting $20,000 per year for administrative costs leaves net social benefits of $20,000 per year. Using the same procedure, it was also indicated that the net benefits could be as high as nearly $2 million per year. Given the data available, it would not be easy to argue against either of these estimates. Each is extreme and the true net gain probably lies somewhere between these two figures.

These figures do need qualification, though, as indicated in Chapter III. The first qualification, and perhaps the most serious, is that when the international arena is considered, the estimates of the net benefits to Canada are too high. Under certain very restrictive and implausible assumptions Canada would, in fact, receive negative benefits from extending copyright protection to computer software and would still have to bear the administrative costs of doing so, leaving Canada with a net loss. These implausibly restrictive assumptions were that Canada would have no effect on the world market for computer software and that there is no national market for software. A relaxation of these assumptions means that the net gains to Canada might be more than -$20,000. How much more and whether they would be positive is impossible to state. The second consideration is the growing market for packaged software and for firmware, both of which will be increasingly costly to protect using trade secrecy since they are produced for wide-scale purchase. The estimates of the possible benefits of copyright protection for computer software did not attempt to include these additional benefits and, in that respect, may have been seriously biased downward. Balancing these two important considerations (international trade flows and future developments in the industry) is difficult be-

cause doing so would require projections into the future using data which are not even available for the present.

The conclusions of the economic analysis of the potential benefits of extending copyright protection to computer software creates an uncomfortable dilemma. The hypothesis that the net benefits are negative cannot be rejected. Nor can the hypothesis that they are positive. The range of the estimates is simply too wide and, unfortunately, spans both positive and negative numbers. Policy prescription requires, however, a yes or no decision, not an estimate of the possible range of benefits. Three observations may help to guide policy prescription on this topic.

First, a serious and perhaps extreme attempt was made to bias downward the estimate of the lower bound of the social benefits of extending copyright protection to computer software. A more plausible lower bound of these estimates is probably considerably greater than $40,000 per year. Second, the administrative costs of the copyright system were considerably overestimated. A more reasonable figure, even allowing for some indivisibilities in the administrative process, is probably less than $10,000. Third, as indicated in Chapter III, it is implausible to assume that conditions in the international software market are such that there would be no gains to Canada from implementing copyright protection of computer software. What portion of the estimated gains would actually exist, however, cannot be determined.

The policy question, then, is this: Should the federal government spend what would probably amount to less than $10,000 per year to obtain the potential gains from making computer programs copyrightable? The answer to this question is yes. While the gains might be less than $10,000 per year, and Canada would suffer a loss, they should, especially in the future, be considerably greater than $10,000. An investment of $10,000 per year in administrative costs would be a good risk for Canada because the potential gain from this investment is great relative to the potential loss.

PART II

COPYRIGHT AND COMPUTER DATA BASES

Chapter I

THE NATURE OF DATA BASES

Introduction

> One problem raised by computer technology is the extent to which the use of copyright material -- for input to a computer or output therefrom -- should be permitted without authorization of the copyright owner, and which of such uses should require permission and the payment of a royalty. The present Canadian Copyright Act does not advert directly to this problem and as such is not amenable to analysis for purposes of finding solutions to the problem. (Keyes and Brunet, 1977, p. 125)

In their provocative study, Copyright in Canada, Keyes and Brunet did not explore each potential issue in great detail. Had they delved more thoroughly into the legal and economic background of data bases and computer technology, they would surely not have made the above statement. It is not at all clear that computer technology in conjunction with information storage and retrieval systems (ISRS) has created any "problem"; it has certainly raised some questions which are addressed in this study, but analysis of these questions does not reveal any serious problems. Equating a question which needs answering with a problem which requires a legislative solution is like trying to find a haystack in which to hide a needle.

As Keyes and Brunet note later in their study, the Copyright Act does state rather explicitly that copyright grants authors the sole right to "reproduce the work...in any material form" (p. 126). This formulation could be interpreted as including material used by or in a computer, as it does not specifically say that the reproduction must be "...perceptible to the [unaided] human senses" (p. 126).

There is no legal problem concerning the use of data bases in ISRS. The next chapter of this study shows that the jurisprudential background, combined with the Act itself, indicates quite clearly what does and what does not constitute infringement of copyright in data bases. Not only is the law clear on this question, but it is consistent with the economic rationales for copyright of data bases.

The economic questions may, however, raise some problems along the lines of detection and enforcement. These problems are not serious and they may be amenable to legislative solution. The economic questions and problems, along with a discussion of possible solutions to the problems, are covered in the next two chapters of this study.

Keyes and Brunet make two major recommendations in their study.[1] The first is that only unauthorized output from an ISRS constitutes infringement, whereas unauthorized input does not constitute infringement. Metaphorically speaking, this recommendation finds a haystack in which the needle can be hidden. It will be argued here that implementation of this recommendation represents a significant departure from present copyright law and would create both legal and economic problems, not solve them. Their second recommendation is that a right of discovery be created to facilitate detection of infringement. This recommendation may have some merit. It will be argued here that the problem of detection may not warrant the right to access a rival's computer data base but may justify the right to examine supporting documentation to determine if the data bases are substantially similar. There are grounds for concern, however, lest the right of discovery become an instrument for competitive harassment or for the discovery of trade secrets.

In fairness to Keyes and Brunet, it should be noted that they intended their study to be a beginning, a provocation of further thought and research; without a doubt, they achieved that goal.

Rationales for Copyright of Computer Data Bases[2]

Data bases, like most literary works, have public good characteristics which give rise to a market failure. Once the data are collected, compiled and published, imitators can expropriate a data base and republish it without bearing the collection and compilation costs. In the absence of copyright protection, the original producers of a data base will attempt to protect their investment by marketing the data base so rapidly that they preempt the bulk of the market from imitators, by charging potential imitators higher prices, by trade secrecy and/or by encryption (i.e., the use of secret codes). Each of these techniques is likely to result in higher costs to the original producer and/or restricted dissemination of the output relative to the costs, prices and outputs under copyright protection. Consequently, society affords such protection to data base producers in the interests of removing or reducing the effects of the market failure, reducing the costs of production and dissemination and encouraging the production of marketable data bases. The invention of the computer and the advances of computer technology do not change this underlying rationale for the protection of data bases. They do, however, change the media in which data bases might be published and they can affect the expected private costs and benefits of the production of infringing copies. In other words,

---

1. They actually make five recommendations but their important recommendations are reduced to two here.

2. There are many thorough discussions of the rationales for copyright protection. See, for example, the introductory section of Chapter II in Part I of this study and the references cited there.

the basis for copyright protection has not changed and will not change; it is only the enforcement of this intellectual property right which might need to be altered in the face of the changing private costs and benefits of breaking the law.

Copyright protection generates social benefits in two ways: by reducing the costs of producing and disseminating copyrightable works. In the first instance, it reduces the costs of protecting the intellectual property of works via rapid marketing, trade secrecy, etc., which would have been produced even in the absence of copyright protection. In the second instance, by reducing protection costs, it encourages the production of additional work. For data bases, copyright confers a benefit on society by reducing the costs of information dissemination. Diagrammatically these benefits are shown in Figure 7. The cost reduction enabled by copyright protection shifts the supply curve down from S to S'. As a result of market forces, the price is then competed downward from P* to P'. Not only will the Q* data bases which had been produced originally now be capable of being produced at lower cost (generating social benefits from more efficient resource use), but also the Q'-Q* data bases which had not formerly been produced will now be produced, generating additional benefits to society. The total benefits of these cost reductions are shown as the shaded trapezoid in the diagram.

For this study, it is accepted that the net benefits from providing copyright protection for data bases are positive. The advent and growth of computer technology do not affect this basic assumption; they do, however, raise some questions concerning the optimal use of society's resources for copyright protection of data bases.

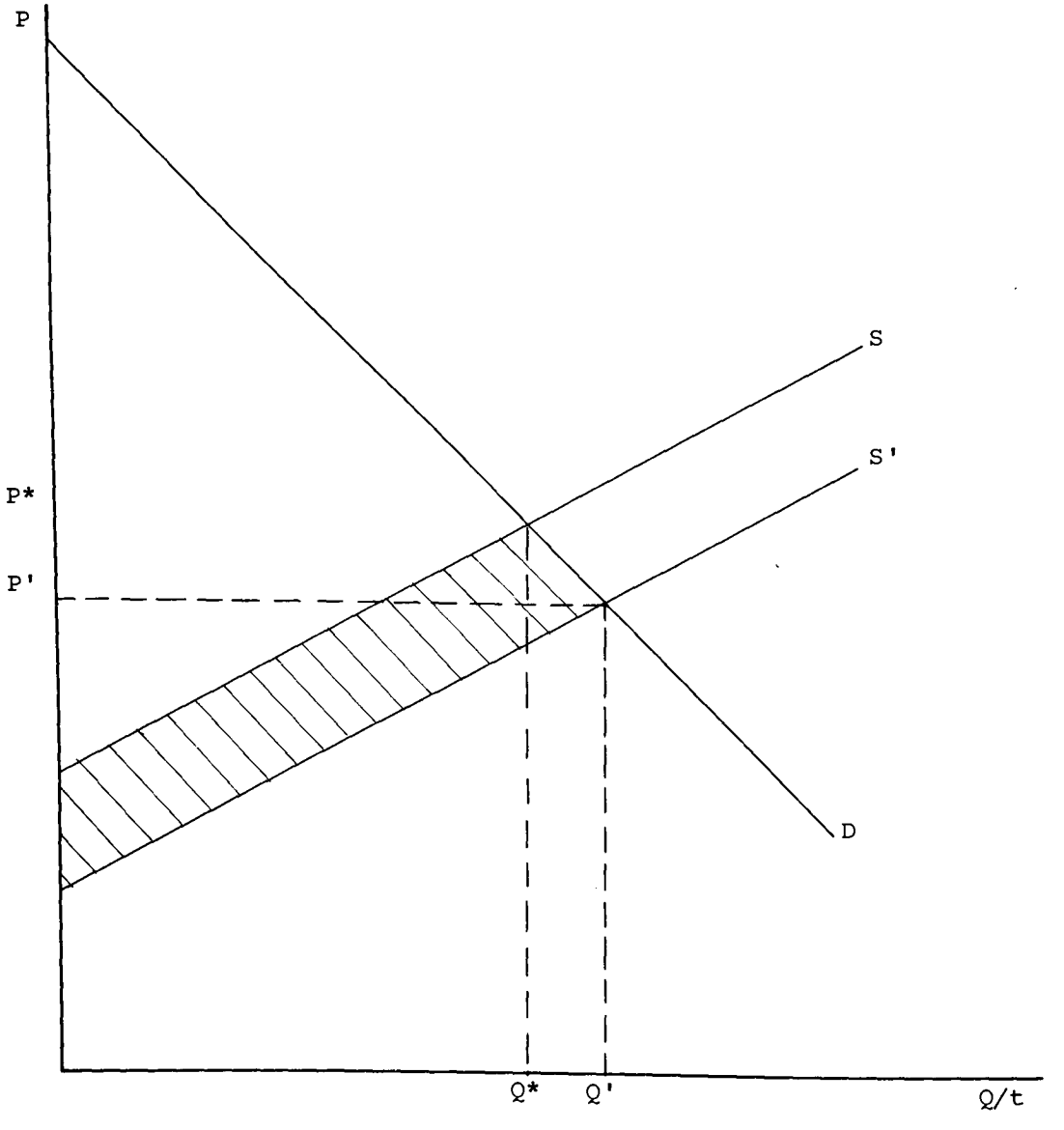
Definitions

In this study, "data bases" will be taken to mean tables and compilations as referred to in Section 2 of the Copyright Act. This section of the legislation presently includes tables and compilations as literary works and hence includes them under the protective umbrella of the copyright form of intellectual property protection. Any particular data base can be thought of as falling into one of four broad categories:

Privately collected for private use. A firm or researcher may collect data not in the public domain for its private use and with no intention of selling the data base to others. Typical examples of data bases falling into this category would be a firm's personnel records or a survey conducted by a firm to determine the impact of a recent marketing program on consumer attitudes. In the latter example, the survey has been conducted to monitor the performance of the marketing division of the firm and to provide data to help the firm improve its future marketing programs. If this process is carried out entirely in-house, the firm usually does not anticipate that there will be any resale value for the data base as an incentive to compile it. In fact, the firm may wish to

Figure 7

Efficiency Gains from New Technology for Computer Data Bases



restrict access to the data base with trade secrecy protection in order to guard against having the data base fall into the hands of its competitors. The firm may, however, see a more generalized value for the data base if it can be used by other divisions within a diversified firm.

Collections of publicly available material for private use. "Publicly available" in this taxonomy means that the material either is in the public domain or is copyrighted by others but readily available, with or without royalty payments, to all who wish to use the material. Many data bases compiled for academic research fall into this category. The data to test a particular hypothesis are gathered from one or more sources and are publicly available. A new data base is often generated using the publicly available materials but, in this category, the new data base has been compiled for private use and not because it has any resale value.[3]

Because the data base in these two categories has been constructed for private use without the ex ante purpose of resale, copyright protection of the data base is not likely to affect the incentives for or the social benefits of its creation. Copyright protection of data bases will affect the incentives and social benefits only to the small extent that some compilations made for private use might unexpectedly have some resale value and so far as such a possibility affects the private calculus of the person or firm making the compilation. Copyright protection will also impinge on the private calculus in these categories insofar as the data bases utilize copyrighted materials for which royalties must be paid.

Private collections for resale. In this category, a person or firm collects material which is not generally available publicly and publishes it for resale. A common example of data bases falling into this category is directories which are compiled by surveying the persons and/or firms to be included in the directories in order to obtain and include information which would not otherwise be readily available to those who purchase the directories. For example, a city directory which lists the names, occupations and employers of each person at each address contains information not available in telephone directories and not readily available through other sources; the material has to be collected privately and compiled with the intention of publication for resale.

Collections of publicly available material for resale. As noted above, publicly available material refers to material in the public domain as well as to publicly available copyrightable material. Data bases in this category are usually compilations of materials which potential customers

---

3. Often, private researchers make their data bases available to others for only the cost of reproducing the data base. In fact, the Journal of Political Economy has a policy (albeit unenforceable) that the authors of work published in the Journal should make their data available at cost to anyone requesting it.

know to be publicly available but which is not in an easily used form. The data are collected and/or selected from one or more sources and reissued to potential customers. Each customer is willing to purchase the material, even though it is available from other sources, because the compiler of the data base is able to make the compilation once and then spread the fixed costs of compilation and publication over many customers, making the published data base cheaper than independent compilation by each customer. This particular category of data bases is becoming increasingly important with the growth of computer technology. Reasonably large data bases can be compiled and recorded on magnetic tape for lease or sale for access by many remote terminals throughout an organization; there are even international computer networks utilizing telecommunication systems to provide access to very large, continuously updated, data bases.

Another distinction between types of data bases which has proven to be of some importance legally is between those involving skill, judgement, competence and/or the "sweat of one's brow," as opposed to those not requiring a substantial input of resources into the collection and/or compilation of the data. In two important U.S. cases, the courts justified upholding copyright protection by emphasizing directly the devotion of labour resources to the tasks of collection and compilation.[4] Because computer-based information storage and retrieval systems can significantly reduce the labour input into compiling data bases, it is clear that the labour theory of value can no longer form a reasonable basis for a test of originality in data bases.[5]

Industry Characteristics

Data bases are a tremendously heterogeneous product. If one could speak of "the data base industry," one would be speaking of products which could have very different characteristics and which could require very different inputs. A data base containing financial statistics about large corporations is very different from a telephone directory. The two are not close substitutes and the cross-price elasticity of demand between the two is low, indicating that the two different data bases are unrelated in demand. Generally, characterizations of an in-

---

4. Leon v. Pacific Telephone and Telegraph Co. (91 F. 2d 484 [9th Cir. 1937]), Jeweler's Circular Publishing Co. v. Keystone Publishing Co. (281 F. 83 (2d Cir.), cert denied 259 US 581 [1922]). For a summary of the U.S. copyright law concerning directories and a discussion of the potential problems created for the law by computer technology, see Oberman, 1974. Canadian case history will be discussed in the next chapter.

5. See also the discussion on originality in the introductory section to Chapter II, Part I.

dustry rely on a definition of boundaries to the industry.[6] These boundaries are determined, in part, by the substitutability in use of the different outputs in the industry; outputs which are reasonably good substitutes for each other are included in the same industry and a break in the chain of substitutes is utilized to help define an industry boundary. So many different data bases are neither substitutes nor complements for each other that they would not, by this standard test, fall into a single industry.

Another standard test to help determine industry boundaries is the cross-price elasticity of supply, which measures the substitutability of inputs rather than outputs. If a firm could easily shift its resources from the production of product A to the production of product B, products A and B may be grouped into the same industry because of these similarities on the input side even if they have no substitutability on the demand side. With respect to data bases, many of these require quite different inputs and have little substitutability on the supply side, while others utilize fairly similar inputs. Contrary to the notion that only semi-skilled labour is required to collect and collate data, different specialized data bases require the employment of labour knowledgeable about the data and about the uses to which it is likely to be put. Nevertheless, in many cases data can be collected by the same employees for quite different uses and the substitutability on the input side is reasonably high.

The unifying theme of interest for this study relies not so much on the traditional definitions of market boundaries based on supply or demand elasticities. Rather, the study will be concerned primarily with those data bases which can economically and feasibly be used in computer-based information storage and retrieval systems. The focus is less on traditional industrial organization questions of market structure and performance than on the role of new technology as it affects the copyright protection of data bases in general, regardless of whether the data bases are in the same industry.

Computers raise questions regarding the copyright protecton of data bases in four different ways. The first is at the authorship stage. Data can be collected and fed into a computer manually but the task of selection and arrangemnt is then left to the computer programmer(s). The computer plays an important role in reducing the labour costs of editing and compiling the data and the programmer(s) or their employers become authors of the data bases for copyright purposes.

---

6. For a discussion of the problems of defining a market or an industry, see Scherer, 1980. The problem of the relevant market has plagued anti-combines legislation (and especially enforcement) since its inception. See, for example, one of the more recent cases in which the definition of the relevant market was at issue, Regina v. K.C. Irving, Ltd. (13 CPR (2d) at 136, N.B.Q.B. [1974]).

A second way involves the stage of putting a data base onto a computer. A data base may be available in printed form but someone wishing to use the data base in conjunction with a computer will find it advantageous to copy the data base into machine-readable form. These uses fall into one of two broad categories: in the first, the data base is used to perform calculations using additional computer software; in the second, it is relayed in small selected sections to a remote terminal, often a cathode ray display unit. In the former use, the data base is of value because it enables the calculation of summary statistics and/or an examination of testable hyotheses about relationships between different parts of the data base (and possibly additional data). In the latter use, the data base is of value because it enables a researcher to scan a large data base quickly to find a specific piece of information, as in an abstract or referencing service.

The third way in which computer technology may affect traditional copyright protection for data bases is in the compilation on the computer of a new data base based on two or more data bases which have already been published. A researcher may wish to examine relationships between variables for which the data are available only from several different sources. To do so, the researcher can write a computer program which selects the required data from each source and then generates a new data base. If this new data base in turn becomes marketable, a potentially difficult problem of assigning copyright royalties is created (see Chapter II on new problems created by ISRS).

The fourth way in which computers affect data bases and copyright involves the outright copying of a data base which is already on, say, computer tapes. In some cases the copying is done simply to maintain a back-up copy of the tape should the first be destroyed inadvertently. In other cases, the copying is done to make the data base available to other computer installations within the organization. And, in still other cases, the copying is done for resale.

This description of data bases and the uses of computers in conjunction with data bases can lead to some interesting questions concerning copyright law, some of which have already been hinted at. The next chapter examines these questions in greater detail.

Chapter II

THE LAW

Jurisprudential Background

The Copyright Act makes it clear that compilations and tables are, for the purposes of copyright, included in the category of "literary works" and are protected as such (s. 2). Consequently, all data bases involving tables, compilations or prose text are copyrightable. The Act further states that: "For the purposes of this Act, 'copyright' means the sole right to produce or reproduce the work or any substantial part thereof in any material form whatever" (s. 3). Thus, with the exception of fair-dealing provisions, the copying of a data base from printed form onto a computer tape or disk would constitute infringement. Similarly, copying a data base from magnetic tape onto a read-only-memory silicon chip would constitute infringement. However, reading data from a computer tape reproduced in selected portions on a video display unit would probably not constitute infringement of copyright because such a reproduction has been held not to be "in any material form" (s. 3).[1]

This brief introduction indicates that Canadian copyright law protects against copying in any material form, whether at the input stage or at the output stage in an information storage and retrieval system (ISRS). Before pursuing the application of the law to ISRS, however, it will be useful to review the history of copyright law as it applies to compilations.

Copyrightable subject matter. In order for a data base to receive copyright protection, it must meet three criteria: it must be original; it must be an expression of an idea rather than an idea expressed; and it must not preempt the use of facts or ideas by anyone else. Each of these criteria will be discussed in turn.

(a) Originality. Copyright is intended to provide an incentive for original work which is likely to benefit society. Because data bases frequently rely on data sources which are available to many different compilers, data bases, even if constructed using independent effort, can end up being very similar to each other. Consequently, the question of the originality of the work is frequently raised. The courts have long held that originality in this instance refers to originality in expression or in effort but not to originality in ideas. Originality in effort seems to be important for establishing the copyrightability of a data base. The courts held in 1874 that copyright cannot subsist in a

---

1. The relevant case is Canadian Admiral Corporation v. Rediffusion Inc. et al. ([1954] Ex. C.R. 382, 14 Fox Pat. C. 114, 20 C.P.R. 75).

work which is already in the public domain.[2] A republication of a data base already in the public domain, hence, does not pass the test of originality. And, in terms of the best use of society's resources, this is a good decision rule; an incentive should be provided to produce new data bases, but not even a limited monopoly should be created and placed in the hands of one person over a data base which already exists. Otherwise, too few resources would be devoted to the production of new data bases and too many would be used to find and copyright already existing data bases.

Originality is often important in copyright cases because a question is raised as to whether a competing data base is original. The courts have held that a competing work is original if it was compiled independently, even if it was based on common sources and had a purpose similar to that of a previous work.[3] There is a clear desire on the part of society and the courts not to stifle independent effort and creativity in the hope that the results, on balance, will benefit society. Allowing the demonstration of independent effort as a defence against a copyright infringement suit is tantamount to establishing independent effort as a criterion for the determination of originality. Because the courts have established this criterion, they have implicitly concluded that the value to society from reducing the costs of protecting creative works and encouraging the production of new works more than compensates for the effect the law has on encouraging a duplication of effort to produce competing and similar works. One of the major benefits of this decision is that potential competition from independently produced works provides a strong check on the exercise of monopoly power by copyright owners. If a copyright owner attempts to raise the price of a copyright work, he must take cognizance of the possibility that competing works could legally be produced and marketed so long as they are produced using independent effort.

The courts have held, from as early as 1887, that independent effort to compile similar material from common sources does not constitute infringement.[4] However, the compiler of a data base must use independent effort to collect and compile the data. He may not use someone else's data base and only verify that it is correct with his independent effort. Instead, he must use independent effort to create his

---

2. Langlois v. Vincent ([1874] 18 L.C. Jur. 160, 2. Can. Com. R. 164).

3. Garland v. Gemmill ([1887] 14 S.C.R. 321, 2 Can. Com. R. 292), affirming Ontario Court of Appeal (unreported) which affirmed 12 O.R. 139.

4. Garland v. Gemmill. See also, Cartwright v. Wharton ([1912] 25 O.L.R. 357, 1 D.L.R. 392).

own data base, but then, without liability for infringement, use another, copyrighted, data base to verify that it is correct.[5]

It has been seen that the law requires originality only in the form of independent effort and not necessarily in the data compiled or in the sources from which the data are collected. Nor does the law require that the originality requirement be interpreted as a requirement of literary merit. As Maclean said in Underwriters' Survey Bureau, Ltd. v. American Home Fire Insurance:

> The Copyright Act makes no requirement as to the value of a literary work; it requires an original literary work; and it is sufficient if there has been labour, skill, time, ingenuity, selection, or mental effort expended in the production of the same. The Copyright Act is not concerned with the originality of ideas, but with the expression of thought in print or writing [emphasis added].

(b) Expressions vs. ideas. A recent case which highlights the copyrightability of expressions rather than ideas is Collins v. Rosenthal ([1974] 14 C.P.R. [2d] 143 [Fed. Ct.]). In this case, two firms had prepared and published similar tax tables. There was evidence of independent effort and there was direct evidence which refuted allegations that slavish copying had taken place. The court emphasized that copyright protection does not extend to ideas but only to their expression. Even if the defendants had copied the idea of computing and publishing tax tables, the fact that they supplied the requisite independent effort to create the tables rebutted whatever presumption of infringement may have existed. Copyright protection cannot be extended to ideas and is available to expressions only if imitations do not expend resources independently on the creation of the expression.[6]

(c) Preemption of facts and ideas. The presentation of evidence of independent effort is important in cases involving the copyright protection of data bases because often a data base may be capable of expression in only a few different ways. Granting protection against independent effort to express a data base in a manner similar to one which has already been copyrighted thus would extend what was intended as a

---

5. The precedent here is found in Emmett v. Meigs ([1921] 1 W.W.R. 35, 16 Alta. L.R. 132, 56 D.L.R. 63 [C.A.]). See also Underwriters' Survey Bureau Ltd. v. American Home Fire Assurance Co. ([1939] 4 D.L.R. 89, Ex. C.R. 296).

6. A major exception to the expression-ideas dichotomy appears in the proscription against unauthorized adaptations. Recent unpublished work by Barry Torno of the Department of Justice suggests that the dichotomy can break down in other areas as well if a "substantial similarity" in works exists.

very limited form of monopoly protection to facts or ideas. The courts have consistently been wary of extending copyright protection to facts or ideas, particularly in areas such as compilations. Yet they are concerned that infringement might take place if the facts in a data base can be expressed in only a limited number of ways. Consequently, a rule has evolved that substantial similarity of the data bases creates a rebuttable presumption that infringement has occurred; the presumption can be refuted by presenting evidence that independent effort was involved.[7] In Canada Bonded Attorney & Legal Directory Ltd. v. Leonard-Parmiter Ltd. ([1918] 42 O.L.R. 141, 42 D.L.R. 342 [C.A.]), the courts held that copyright protection to a list of bonded attorneys does not preempt the use of common sources by others to compile a similar list. The courts would not extend the copyright monopoly to the idea of producing a list; nor would they extend the protection to the expression of the list if the competing expression were compiled independently. They clearly wished to limit the scope of copyright protection to a very small area and not to allow the preemption of facts or ideas with a copyright monopoly. But similarity of published data bases does create a rebuttable presumption of infringement. In Emmet v. Meigs, the courts held that infringement had occurred because the defendant offered no satisfactory explanation of the similarities in the published works. In order to avoid copyrighted monopoly control of facts, similar data bases must be allowed. But to avoid rampant infringement, similarity must be accompanied by a demonstration that independent effort went into the compilation of the competing data base.

While unsatisfactorily explained similarities may be grounds for finding that an infringement of copyright has taken place, such a finding usually hinges on the meaning of "unsatisfactory." The interpretation is left to the courts to decide on a case-by-case basis. Far easier for the finding of infringement is the existence of common omissions and/or common errors in two competing data bases.[8] In Cadieux v. Beauchemin, the courts held that the publication of a dictionary containing many errors in common with those in an earlier published dictionary would constitute a prima facie case of infringement. More re-

---

7. Underwriters' Survey Bureau Ltd. v. American Home Fire Assurance Co.; Beauchemin v. Cadieux ([1900] 2 Can. Com. R. 337 at 348, 10 Que. K.B. 255, 31 S.C.R. 370, 2 Can. Com. R. 170). While this rule seems to have held for compilations, it may not readily be generalized to other areas. See footnote 6.

8. When a data base is compiled independently, it is reasonable to expect that it would have many similarities to other copyrighted data bases containing the same information. It is extremely unlikely, however, that it would have errors, misprints and omissions exactly the same as those in other data bases; the probability of making similar mistakes is very low whereas the probability of similar correct inclusions is very high. For a discussion of this issue, although it was tangential at best in the case, see Deeks v. Wells ([1933] 1 D.L.R.

cently it was held in Latour v. Cyr that the repetition of omissions, misprints, and errors in a second directory and which appeared identically in the plaintiff's directory was evidence of infringement ([1951] 11 Fox Pat. C. 136, Ex. C.R. 92, 15 C.P.R. 21).

The distinction between the criteria for copyright protection of data bases -- originality, expression, no preemption -- are blurred because the rationale for the criteria is single-minded. Society wishes to encourage the production of valuable data bases by restricting unauthorized appropriation of them but, at the same time, does not want to restrict the flow or generation of ideas or the publication of facts. Consequently, a relatively simple rule has evolved: individuals may recreate data bases published and copyrighted by others so long as they do so with an independent expenditure of resources. The three criteria discussed are different but not mutually exclusive approaches to the application of this simple rule.

New Problems Created by ISRS

What is a copy? In the introduction to this chapter, it was indicated that the courts probably would not hold the line-by-line reproduction of a data base on a video display unit to be an infringement of copyright.[9] One question facing the courts if the Act remains unchanged from its phrase "...in any material form whatever...[emphasis added]" is what is meant by "material form." The courts in Canadian Admiral applied a test of durability. The test of "reasonable substance or permanence" was given strength in earlier copyright cases involving news stories and was applied by analogy in Canadian Admiral.[10] Extending these applications of the law, because magnetic tapes, floppy disks or silicon chips have reasonable substance and permanence, copying a data base onto one of these media would, under the present Act, constitute infringement. And copying the data, line by line, onto a sheet of paper using a computer printer would also constitute infringement even if that copy were used in precisely the same manner as a fleeting line-by-line copy on a video display unit and were subsequently de-

---

353), affirming (1931) O.R. 818, 4 D.L.R. 533, which affirmed (1930) 4 D.L.R. 513 (P.C.).

As mentioned earlier, the value of the common mistake as evidence in infringement suits has led many firms producing data bases to include purposely some mistakes in their data bases so that a presumption of infringement can easily be established should an infringement occur.

9. Canadian Admiral v. Rediffusion.

10. See also Fox, 1967, p. 102.

stroyed.[11] One question to be addressed in the next chapter is whether, given what the law defines to be a copy, Canada should change the definition to respond to new technology.

Copyright at the input or output stage? As the law reads now, with the exception of fleeting, nondurable reproductions, copyright protection for data bases is available at both the input and the output stages. Copyrighted sources may not be used to produce unauthorized magnetic tapes of data bases. Nor may copyrighted magnetic tapes be used to produce computer printouts or other copies in a durable material form. Keyes and Brunet have argued that copyright protection should not include protection at the input stage, subject to a right of discovery (1977, pp. 128-129). They would allow economic agents to copy, without permission, copyrighted data into an ISRS. Their recommendation is based on the consideration that speed is often extremely important in the information processing industry and that negotiations over royalties may impose socially costly delays to users. They would allow users to input data with impunity and to utilize the data for information processing purposes but they would not allow unauthorized output of the data.

It is interesting that the Keyes-Brunet recommendations concerning copyright protection and computer-based ISRS have made a distinction between the input and the output of data bases. This distinction represents an important break with the legal history of copyright, which clearly grants copyright protection against all unauthorized copying (in material form) regardless of intermediate technologies. In fact, the Act, with respect to damages, grants possession of infringing plates for making copies of books to the copyright owner (s. 21). This provision for the award of damages makes it very plain to the careful reader of the law that traditional copyright protection has long been available at both the input stage (the infringing plates) and the output stage (the infringing copies). A second question, then, to be addressed in the next chapter is whether computer technology has sufficiently altered the possible market failure in the production of data bases tojustify abridging this feature of the Copyright Act so that copyright protection is granted at only one stage rather than at all stages of production.

---

11. The evolution of this apparent inconsistency in the law has a rational basis. Printed copies can be resold much more easily than fleeting images on video display units, regardless of whether they actually are sold. Because of this greater probability of reducing the returns to the producers of data bases, printouts must be authorized while video displays need not be. The decision rule is not perfect since some printouts would not be used to infringe the producer's rights and some video units could be used to generate infringing copies. Nevertheless, the relative expected social costs of generating infringing copies with the two technologies are likely to be sufficiently different that prohibiting one while permitting the other is defensible.

Monitoring. Recent literature on copyright and new technology has pointed out that if a data base producer (or any other potential copyright owner for that matter) is able to practise price discrimination among customers based on the number of reproductions which each customer will make, then copyright protection would not substantially alter the allocation of resources because the creator would be able to appropriate all or at least much of the rewards for the creative effort and will receive the desired incentives to create original works. In the case of centralized ISRS, in which access to data bases is sold but the data bases themselves are not sold, price discrimination by use is facilitated, as is reflected by the various marketing schemes of data base suppliers. These schemes often include a fee calculated from the number of times the data base is accessed in the ISRS and the cumulative length of time that the data base is accessed. Despite the prevalence of these types of price discrimination schemes, they are often not directly related to the recovery of revenues from unauthorized copying because they most often are implemented in conjunction with video display units from which unauthorized copies would rarely, if ever, be reconstructed. The schemes do provide a means for capturing rewards in circumstances in which subscription to an ISRS data base is done in lieu of purchasing a number of printed copies for use throughout an organization. In these circumstances, subscription fees will be positively correlated with the number of printed copies which would have been purchases were no ISRS data base available. Such monitoring can also be negotiated at the input stage for data bases which are purchased outright. The producer of a data base can sell a computer tape of the data base to an organization along with a "service" contract which specifies that the organization must keep accurate records of the use of the data base and which specifies the sale price as a fixed payment plus a royalty based on the use of the data base. These possibilities indicate that there are numerous market options for monitoring the use of copyrighted data bases.

The detection of infringement. The detection of infringement, whether it be a breach of contract as discussed above or infringement of copyright or both, can be difficult with computer-based data bases. In the more flagrant possibilities, such as reproduction of a data base tape for resale, detection of infringement may be easier if a data base producer is unable to sell the data base to potential clients because they have already purchased pirated copies. Even in these cases, legal proof that the potential clients have a pirated copy and acceptable evidence indicating who produced the infringing copy may be difficult to acquire. For these reasons, Keyes and Brunet have recommended that a discovery right be included in the Copyright Act to allow a data base producer access to another economic agent's records and even its data base to determine if and by whom an infringement has occurred. The proposed discovery right has merit to the extent that it increases the likelihood that a data base producer will be able to appropriate the value of creative effort and, by doing so, it provides additional incentives to produce new data bases. Potential problems could be created with a discovery right, however, to the extent that it could be used by

one firm to acquire the trade secrets of another firm or that it could be used illegally to upgrade the firm's own data base with better, more complete, more reliable or more recent data.

Infringement for the purposes of resale may seem more flagrant than other types of infringement but economically the effects are similar. A client purchasing a data base and reproducing it for use in other branches of its own organization is depriving the data base producer of revenues just as much as if it purchased pirated copies of the data base. Any reproduction, even for private use, has the same effect as the purchase of an unauthorized copy from another person or firm: it serves the (short-run) socially useful function of disseminating the data base more widely than would have occurred if the data base could be purchased only from the producer at a higher price but it reduces the returns to the producer each time an infringing copy is purchased or produced for private use when an authorized copy would otherwise have been purchased or produced. Whether an infringing copy is made for resale or for private use, the problem of detection may seriously exacerbate the market failure which copyright protection is designed to remove or reduce. The question of whether additional legal protection is necessary to facilitate the detection of infringement is the third question which must be addressed in the next chapter.

Private copies and fair dealing provisions. Very large, continuously updated data bases accessed from remote terminals through computer networks are not likely to be copied either for private use or for resale because the costs of copying and updating (and marketing) them are greater than the expected private benefits. The problem of private copies is more relevant for smaller, but especially more durable, data bases which are typically sold either as printed copy or on magnetic tapes. Purchasers of these data bases may wish to make private copies for one or more reasons. They may simply wish to have a backup copy of the data base should one become inadvertently destroyed. They may want to make copies for use elsewhere in their organizations. Or they may find that they can make more efficient use of the data base if it is transcribed into a different medium.[12]

Recall that a primary motivation for copyright protection of data bases is the promotion of dynamic economic efficiency. Impediments to the private use of data bases may thwart society's pursuit of this goal. Yet allowing free use of copyrighted data bases may just as easily impede pursuit of this goal by reducing the incentives to produce data

---

12. For example, if the data base is available in only printed form a client may find that purchasing several copies of the data base or photocopying relevant sections of the data base would be less efficient than having the data base on computer tape for access from a number of remote terminals. More simply, a client may find that having the data base on magnetic tape greatly reduces the costs of information processing.

bases. To resolve this dilemma, Keyes and Brunet have recommended that the input of data bases, regardless of their use, essentially be allowed a fair-dealing defence as not constituting infringement. The implicit decision in their recommendation is that the value to society of allowing unauthorized input of data bases into ISRS for information processing purposes more than outweighs the deleterious effects that doing so would have on the incentive to produce data bases capable of use in an ISRS. This recommendation significantly alters the present concept of fair dealing by allowing unauthorized input for commercial uses as well as for private uses. As has already been demonstrated, it represents a departure from traditional copyright jurisprudence and, as will be discussed in the next chapter, it is not clear that such a departure is warranted by the new technology.

Rearrangements, combinations and syntheses. One of the great labour-saving features of computer technology is the speed and relatively low cost with which data can be sorted, combined and/or selected for processing and possible reproduction. This section discusses the legal questions and the potential economic problems which might be raised by the ease with which computers can be used in conjunction with an ISRS to produce data bases for resale.

(a) Rearrangements. Every reasonably well-skilled computer programmer can write a program which will sort and resort data and then print out or store on magnetic tape the new arrangement of the original data base. While these new arrangements of the data might defensibly be viewed as new expressions of an idea, they have tended to be viewed by the courts as more analogous to translations and adaptations and, hence, to be infringing copies.[13] The judgements in the cases have emphasized that, although a rearranged data base may be of additional value to society beyond the value of the original arrangement, because rearrangement per se does not require independent utilization of resources to collect the data, it does not qualify as an independent creation.[14] The courts have, in effect, decided that where there are private (and social) benefits to be generated from rearranging an existing data base, there will be gains to the private parties from negotiating a contract for either the original producer of the data base or for her contracting agent to produce the rearranged data base as well. The courts have generally been correct in this decision, insofar as allocative economic efficiency is concerned, so long as the negotiation and transaction costs of contracting for the production of a rearranged data base are low. If, because of personality clashes or strategic bargaining decisions, these costs are high, then the private and social gains will be

---

13. A classic example of this situation is provided by the U.S. law: Leon v. Pacific Telephone and Telegraph Co. ([1937] 34 USPQ 237).

14. Leon v. Pacific Telephone and Telegraph Co. and Underwriters' Survey Bureau Ltd. v. American Home Fire Assurance Co. ([1939] 4 D.L.R. 89, Ex. C.R. 296).

eroded and, in these cases, society will not benefit from the decisions which have made arrangements of data bases an infringement.

(b) Combinations. Different and independently produced data bases may be of sufficient value to society that there is a market for a combination of them. Current legislation would consider the combination an infringement of the individual copyright because the combination would be a reproduction of each of the original compilations. The rationale for such a policy is, presumably, the same as that for rearrangements: if there are private and social gains to be had from combinations, the privately contracting parties will seek the gains through private negotiations. The problem is potentially more serious with combinations of existing data bases, however, because more individuals are party to the bargaining process, each seeking as large as possible a portion of the private gains. Consequently, the negotiation and transactions costs will most likely be greater, increasing perhaps expotentially rather than linearly with the number of parties included in the bargaining. Whether the costs increase exponentially or linearly is of less concern than that they will increase monotonically with the addition of more parties to the negotiation process, thus reducing the net social benefits of relying on the marketplace to generate data base combinations.

(c) Syntheses. If computer technology creates an economic efficiency quandry for rearrangements and combinations of data bases, the problem is potentially even greater for data bases which are reproduced selections from several different copyrighted data bases. Not only are the efficiency problems discussed above at issue but so is the legal question of what constitutes "a substantial portion thereof." Selecting 50 per cent from each of two data bases and producing a synthesis of the two would probably constitute infringement of each copyright.[15] The courts, in the absence of legislative direction, may soon be called upon to decide if selecting ten per cent from each of ten different data bases is infringement. And what about selection and synthesis of 0.1 per cent from each of a thousand different data bases? What is and what should be the decision rule in such cases?

Even assuming that the courts can decide what constitutes infringement reasonably well on a case-by-case basis, in cases for which royalties would or could be required the bargaining process may create even greater costs than would be created in the circumstances described for pure combinations of existing copyrighted data bases. Not only might more parties be involved in the negotiations but the relative amount of each original data base to be reproduced in the synthesis will also be at issue. Each copyright owner would have an incentive to hold out for a large share of the gains if the synthesis cannot proceed with-

---

15. Some guidance on this question might be available from Cartwright v. Wharton ([1912] 25 O.L.R. 357, 1 D.L.R. 392), but not much.

out his or her authorization.[16] Allowing such issues to be adjudicated in the court system may eventually provide sufficient guidance for individual copyright owners that economic efficiency will obtain. Yet, a copyright tribunal, an arbitration board or more specific legislation in this area may be more efficient even in the long run.

The potential economic efficiency and legal questions raised by rearrangements, combinations and syntheses of copyrighted data bases are intensified by the great potential for cost reductions made possible for these products by computer technology in an ISRS. Suggestions for dealing with these issues and the others discussed in this chapter will be presented in Chapter III.

---

16. Many of these issues exist for combinations as well. It should be noted that there does not appear to be any case law in Canada dealing specifically with the issues raised here.

Chapter III

RECOMMENDATIONS

Generally speaking, the Copyright Act provides adequate protection for data bases used in computer-oriented ISRS. In anticipation of some relatively minor potential problems in the future, Parliament should, however, consider making some changes in the Act, primarily in areas which will speed information flows in the economy and ensure that data base producers will receive sufficient rewards to induce them to continue to produce data bases for use in ISRS. The discussion in this chapter follows the same broad outline of the questions raised in the last chapter, with each question discussed in turn.

What is a Copy?

As the result of some undefinable combination of luck, Parliamentary foresight, judicial prudence and evolutionary economic efficiency, a very good definition of what constitutes a copy of a data base presently exists in the law. Hard copies of some permanence and substance are considered copies, while copies which are impermanent, such as video displays, are not considered copies for the purposes of the Act. This definition gives the data base producer control over copies which could otherwise reduce his or her revenues and does not restrict copies which would generally not reduce producer revenues.

Fleeting images on video display units are not likely to reduce producer revenues because the costs of handcopying the data from the screens and reproducing them would generally be greater than simply purchasing the data base. At first blush, it might seem that allowing a number of video units to utilize one computer tape might reduce revenues from the sales of the tape or of the printed form of the data base. This is not necessarily correct. Under the Act, no one can make a computer tape of a data base without authorization from the copyright owner. Eventually, in fact, the copyright owner will discover that it is cheaper for him to reproduce and market the data base on tape than it is for each customer to produce a tape from the printed form. Consequently, society will gain from the lower average cost of producing the data bases on tape. An intriguing marketing problem faces the producer, though. If he markets only the tape, he will lose sales to those customers who do not have an ISRS available. Furthermore, if he sells it at too low a price, he will lose profits from the sales of printed copies to formerly multiple purchasers who now purchase just one copy of the data base on magnetic tape for use at numerous remote terminals. Because the production of a data base in either tape or printout form involves few fixed costs beyond the initial collection and compilation costs, the producer in this case will usually continue to market the printed copies to clients without an ISRS and will market the tapes at a higher price, perhaps with an access fee, to the customer using ISRS. By producing the data base in two different media, the producer can most likely avoid running afoul of the Com-

bines Investigation Act, section 34 of which proscribes price discrimination between competitors, and still be able to capture some of the gains from the new technology by effectively practising second degree price discrimination.[1]

Consider three different possibilities. In the first possibility, assume that no infringement will take place. If producing the data base in both media is less profitable at any set of prices than producing it in only one media, the producer will select only the most profitable medium for production. In the second possibility, assume that every potential customer for whom self-production of an infringing tape is profitable could and would do so without detection. Under this assumption, the data base producer would be better off to sell the tapes himself at a price marginally below the costs incurred by self-producing infringers and at least make higher profits than otherwise from these potential infringers. In the third possibility, assume that only some infringement would take place. Again, giving the producer of the data base the option of practising at least partial price discrimination can only make the producer better off. In general, increasing the set of options open to a producer can never make the producer worse off and that is exactly what happens when the producer is faced with the possibility of practising price discrimination by selling both computer tapes and printed copies of data bases.[2]

The purpose of this fairly long discussion is to show that the present definition of a copy is adequate, even with the advent of computer-based ISRS. Fleeting images on video display units represent little threat to socially desirable incentives to create data bases because they cannot easily be reproduced for resale and because price discrimination can be employed by the data base producer to appropriate a portion of the gains resulting from using the data base in conjunction with an ISRS.

Input vs. Output

The major argument for allowing input of data bases into ISRS without liability of infringement is a fear that protracted negotiations over authorization and royalty payments will impede the socially beneficial flow of information. There is a justifiable concern that conscientious researchers will be hampered in their work if they must first negotiate authorization to input data into an ISRS. It is further argued that allowing unauthorized input would not seriously affect incentives for the production of data. This argument has force primarily for data

---

1. The economic theory of this argument is complex. See Liebowitz, 1980(a).

2. This discussion has completely ignored the possibility of more refined price discrimination which can be practised if the amount of use of a computer data base can be monitored.

bases used for information processing and from which no copies are made. For all data bases which are also outputted, there is little difference between creating a property right at the input stage or at the output stage since negotiations over royalties must take place eventually.

Given the arguments made earlier that video display images not be considered copies, allowing free or unauthorized input of data bases into an ISRS might impinge upon the expected revenues of the producers of data bases capable of being used with video display units. Practically speaking, it may not, with current technology, make much difference because material would normally have to be inputted into an ISRS with punched cards or by making a disk or tape of the material for later use at the remote terminals. Doing so, at least under current law and under the recommendations put forward here could, except possibly in cases of private use, constitute an infringement if it were unauthorized. However, Keyes and Brunet seem to be recommending that such activities be exempt from copyright protection. If so, their recommendation will reduce expected revenues for all potential producers of data bases susceptible to such unauthorized use and, consequently, reduce the incentives for producing these data bases. With the potential for continued growth of computer networks within organizations, this reduced incentive to create data bases may well be more serious than the possible delays in information flows resulting from impeded bargaining over royalties at the input stage.

The distinction between input and output of computer data bases is unfortunate. Not only is it of dubious value in projecting social benefits but it represents a significant divergence from traditional copyright law, which protects all copies in material form regardless of whether they are inputs or outputs from any process.

The Problem of Detection

Because data bases used in an ISRS are not readily identifiable by the unaided human eye and because copying a relatively large data base onto a second magnetic tape is relatively inexpensive, there is some concern that computer technology will promote an increasing incidence of undetected infringements of copyright in data bases, thus decreasing both the expected rewards to data base producers and the incentive to create additional information for society. Keyes and Brunet have therefore recommended that a discovery right be created so that data base producers can more easily determine whether another data base is an infringing copy. They suggest that a fairly restrictive set of conditions be satisfied before the producer could gain access to the actual allegedly infringing copy. By recommending this restrictive set of conditions, Keyes and Brunet are responding, at least implicitly, to concerns that the right of discovery might become a legal tactic for harassing competitors or acquiring their trade secrets.

Their recommendations could be improved toward this end with only slight modifications. The party exercising the right to discovery should be required to demonstrate that it has exhausted all other legal channels and that it has good reason to believe that the potential defendant's data base is an infringing copy. Additionally, the plaintiff should not initially gain access to the defendant's data base. The question of similarity should be decided first by comparison of supporting documentation and second by the courts or a copyright tribunal, possibly with the aid of court-appointed or tribunal-appointed arbiters. This policy would reduce the likelihood that an innocent defendant would be forced to divulge trade secrets to the plaintiff. It could be costly, however, because it could take several days for an expert to compare the data bases and write a report; a reasonable cost figure might well be in the range of two to four thousand dollars at 1981 prices. To inhibit frivolous uses of the right of discovery and to reduce its use as a tactic for harassment, the plaintiff should be required to post a bond in advance equal to the expected costs of examining the two data bases. If it is determined that the data bases are sufficiently similar that the plaintiff has a case, the bond should be returned, the plaintiff should gain access to the data base of the defendant and the defendant should be required to pay the costs. If the data bases are sufficiently dissimilar that an infringement suit would be dismissed, the plaintiff's bond should be forfeited.

Clearly, there are social costs to this scheme, including the value of society's resources used in preparing, defending, examining and hearing the case. In fact, these costs would probably be greater than the benefits for any particular case. Since most of the costs would be borne privately, they would, however, provide an incentive for the disputing parties to reach an agreement before the right of discovery was actually exercised. There would still be some scope for strategic behaviour (i.e., threatened use of the right of discovery to intimidate one's rivals), but the scheme proposed here would tend to mitigate this effect, at least relative to that proposed by Keyes and Brunet. Based on this discussion, the social benefits of having a right of discovery are questionable. If they are positive, this right must be limited in order to restrict undesirable trade practices.

A major drawback to these recommendations is that the courts or tribunal would be making a finding of similarity or dissimilarity based on some evidence which the plaintiff would not be able to examine prior to the finding. To alleviate the possible deleterious effects of this potential bias against the plaintiff, the courts or tribunal should accept fairly lengthy and careful direction from both parties, but especially the plaintiff, as to what to look for as evidence of similarity or dissimilarity. They should also be directed legislatively that the standard of proof would be lower in a hearing on the right of discovery than in an infringement suit.

If a copyright tribunal or an expanded Copyright Appeal Board is created in a revised Copyright Act, it would perhaps be more appro-

priate to have the right of discovery referred by the courts to the tribunal for a decision concerning the similarity of data bases. The tribunal could then make a decision analogous to an indictment in criminal cases -- namely, that there is sufficient evidence of similarity in the data bases to justify allowing the plaintiff access to the defendant's data base to prepare an infringement suit. The tribunal could be required by law to set a fee (penalty) which would at least recover its fully allocated costs; and the finding of the tribunal should not necessarily bind the courts.

Fair-Dealing Defence

As has been argued elsewhere with regard to copyright law and computer technology, the fair-dealing defence must be construed very carefully. To date, the courts have made decisions in agreement with this recommendation so far as compilations are concerned (noted in the last chapter). Fair-dealing defences have been allowed if a copy was made to study what data are included in a data base or for research purposes to cross-check and verify the results of independent effort. Other unauthorized copies made, albeit possibly for research purposes but to use the data to study another issue rather than to study the data base per se, might be considered infringing copies, depending on the courts' interpretation of the law.

Academics, among others, probably want to use data bases free of royalty payments or purchase prices solely for information processing purposes and not for reproduction and resale. Allowing them to do so could erode the revenues of individuals and firms producing data bases for such uses to some minor extent and reduce the incentive to produce such data bases in the future. Fair-dealing provisions have been included in the law, however, because the (implicit) general consensus is that the additional benefits of requiring authorization for copies used in private research would not outweigh the additional transaction costs. This view is consistent with the economic rationales for copyright protection. The existence of present fair-dealing provisions and the rationale for their existence seriously negate any arguments for allowing all unauthorized input of material into an ISRS. Roughly speaking, under current law, unauthorized input would not violate copyright if it were for private research but would if it were for commercial use. While the distinction between private and commercial use may be fuzzy in some instances, it is a distinction which is evolving through judicial interpretation. Furthermore, it is a distinction which will be no more fuzzy with the advent of computer technology. In most cases, the potential "problem" to which Keyes and Brunet advert will not exist under the current Act because private researchers would not be liable for copyright infringement if they copied data bases into an ISRS for their own use. Replacing this limited form of fair-dealing provision with the very broad permission which would allow unauthorized input regardless of use is bad policy for two reasons: first, it is not generally necessary since the more limited form of fair dealing is adequate (i.e., it would confer no benefits on society); and second, it would be

costly in that it could have a detrimental effect on the incentives of data base producers.

Rearrangements, Combinations and Syntheses

(a) Rearrangements. It was indicated in the last chapter that rearrangements of copyrighted compilations have in the past been found to be infringements. In keeping with prior arguments and the basic rationales for copyright protection of compilations, it is reasonable to continue to find unauthorized rearrangements produced for resale to be infringing copies since they make considerable use of the original effort to collect and compile the data. Allowing rearrangements, even produced with the aid of an ISRS, would reduce incentives to devote resources to the original effort. In keeping with the above arguments for fair-dealing provisions, however, rearrangements for private use should not be proscribed.

(b) Combinations and syntheses. Combinations and syntheses present similar economic problems. Data base syntheses may be more complex because of the legal uncertainty of determining what is a substantial portion of a previously copyrighted data base but the economic problem is one of reducing transactions costs so that socially valuable combinations and syntheses will be produced. The current provisions in sections 13 and 14 of the Copyright Act for compulsory licensing may be inadequate to reduce these transactions costs. They do not allow for the possibility of a breakdown in the bargaining process to produce a data base combination or synthesis. The provisions for compulsory licensing of sound recordings are equally inadequate; they allow only two cents per playing surface as a royalty.

Some form of compulsory licensing scheme might reduce bargaining costs and facilitate the production and flow of information but structuring such a scheme could prove difficult. Certainly the royalties should not be fixed in the law, even in percentage terms, since price-cost margins vary tremendously with expected sales volumes. Consequently, the only form compulsory licensing could reasonably take would involve arbitration by a tribunal or in the judicial system. Such arbitration also involves social costs and time delays, however, and perhaps it could only be done by the copyright tribunal recommended by Keyes and Brunet (1977, p. 223).

One of the major reasons for delay and drawn-out bargaining is that each party is willing to devote time and resources to the negotiation process in order to get a better deal. Those who advance arguments favouring compulsory licensing and expressing concern for delays in information dissemination are generally data base users attempting to enlist government support in their strivings for a better deal. They question whether data base producers should be able to hold out for high royalties when society would benefit from a speedier diffusion of

information.[3] This argument ignores the long-run signaling effects of high monopoly profits. It is more sensible for society to put up with the potential problems of short-run impediments to information dissemination in favour of long-run incentives to produce more information than it is to require that a copyright tribunal make case-by-case arbitrations when bargaining breaks down. Allowing for arbitration of royalties by the federal government would, in fact, create an incentive for parties to hold out in hope of getting a better deal from the tribunal. As a result, the time costs of bargaining and arbitration could well be greater than if compulsory licensing were not available beyond what is already provided by statute.

Conclusion

Data bases produced for use in information storage and retrieval systems present no legal problems under the current Copyright Act. It has been argued at length in Chapter III that no new definition of what constitutes a copy is necessary, nor does the distinction between input and output stages serve any practical purpose. Furthermore, the present definitions and current legal practice are in accord with what economic rationales for intellectual property suggest should be the state of affairs. The only potential problems emerge because computer technology greatly reduces the costs of producing infringing copies of data bases. The pros and cons of legislating a right of discovery to facilitate detection of infringements were considered and extreme caution is urged if such a right is implemented. Similarly, caution is urged in structuring a compulsory licensing scheme and the desirability of any such scheme is open to serious challenge.

In conclusion, this study has shown that the status quo copyright protection for data bases is also probably the optimal form of protection even in the computer era. It provides an inducement for the creation of socially valuable data bases yet it limits this inducement by allowing for independent effort to produce competing data bases. In sum, the status quo creates a relatively minor change in market conditions to correct for a market failure. Nevertheless, it places a great reliance on the market to allocate resources efficiently. It is difficult to recommend that this reliance on the market will not be optimal in the future as well.

---

3. For analogous arguments concerning cable television see Keyes and Brunet, 1977, pp. 140 and 144. They did not, however, recommend compulsory licensing for ISRS.

Appendix A

STATISTICS CANADA DATA

The data for the industry study based on Statistics Canada publications are taken from the Computer Services Survey (cat. no. 63-222) from 1972-1978. The first two years of the survey included firms primarily engaged in hardware rental along with other firms in the aggregate data reported. The published report for 1974 did not allow for a breakdown by size classes. And, in 1978, the survey was shortened so that less data were collected and reported. Also, it appears from the Statistics Canada discussion of the collection of the data that more firms were reached and surveyed in each succeeding year. The data used to compile the tables discussed in the text are presented here as Tables A1-A6. The size class boundaries are by 1000s of dollars in total operating revenues. Service operating revenues are also recorded in 1000s of dollars. An "X" indicates unreported data in compliance with confidentiality restrictions placed on Statistics Canada.

Table A1

1972					
size classes	less than 100	100-499	500-1999	more than 1999	total
no. of establishments	147	95	46	21	309
no. of employees	533	1353	2431	10097	14414
service operating revenues	6171	21150	42799	80135	150255
software and systems service revenues	1595	3856	8833	13191	27475
software package revenues	433	1298	3454	8049	13234

Table A2

1973					
size classes	less than 100	100-499	500-1999	more than 1999	total
no. of establishments	142	112	47	30	331
no. of employees	575	1582	2066	9950	14173
service operating revenues	5471	23112	38113	108792	175488
software and systems service revenues	1366	5714	5936	23048	36064
software package revenues	571	1304	2132	7398	11405

Table A3

1975

size classes	less than 100	100-499	500-1999	more than 1999	total
no. of establishments	144	160	63	30	397
no. of employees	445	1869	2810	4598	9722
service operating revenues	6553	40012	61428	177688	285681
software and systems service revenues	2870	11252	14603	23275	52000
software package revenues	X	X	X	X	12717

Table A4

1976

size classes	less than 100	100-499	500-1999	more than 1999	total
no. of establishments	198	179	79	33	489
no. of employees	570	1885	3047	4779	10281
service operating revenues	9102	42148	74374	201907	327531
software and systems service revenues	4221	11701	18210	25439	59571
software package revenues	X	X	X	X	11185

Table A5

	1977				
size classes	less than 100	100-499	500-1999	more than 1999	total
no. of establishments	246	215	88	47	596
no. of employees	575	2087	3111	6106	11879
service operating revenues	10696	51128	83760	270464	416048
software and systems service revenues	5194	15271	16880	40060	77405
software package revenues	X	X	X	X	15302

Table A6

	1978				
size classes	less than 100	100-499	500-1999	more than 1999	total
no. of establishments	282	256	108	52	698
no. of employees	629	2088	3374	7105	13196
service operating revenues	12248	60262	103456	355820	531786
software and systems service revenues	X	18827	26386	X	112150
software package revenues	X	X	X	X	20933

Appendix B

MODEL PROVISIONS FOR SOFTWARE PROTECTION (WIPO)

Section 1

Definitions

For the purposes of this Law:

(i) "computer program" means 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;

(ii) "program description" means a complete procedural presentation in verbal, schematic or other form, in sufficient detail to determine a set of instructions constituting a corresponding computer program;

(iii) "supporting material" means any material, other than a computer program or a program description, created for aiding the understanding or application of a computer program, for example problem descriptions and user instructions;

(iv) "computer software" means any or several of the items referred to in (i) to (iii);

(v) "proprietor" means the person, including a legal entity, to whom the rights under this Law belong according to Secton 2(1), or his successor in title according to Section 2(2).

Section 2

Proprietorship; Transfer and Devolution of Rights in Respect of Computer Softwares

(1) The rights under this Law in respect of computer software shall belong to the person who created such software; however, where the software was created by an employee in the course of performing his duties as employee, the said rights shall, unless otherwise agreed, belong to the employer.

(2) The rights under this Law in respect of computer software may be transferred, in whole or in part, by contract. Upon the death of the proprietor, the said rights shall devolve according to the law of testamentary or intestate succession, as the cae may be.

Section 3

Originality

This Law applies only to computer software which is original in the sense that it is the result of its creator's own intellectual effort.

Section 4

Concepts

The rights under this Law shall not extend to the concepts on which the computer software is based.

Section 5

Rights of the Proprietor

The proprietor shall have the right to prevent any person from:

(i) disclosing the computer software or facilitating its disclosure to any person before it is made accessible to the public with the consent of the proprietor;

(ii) allowing or facilitating access by any person to any object storing or reproducing the computer software, before the computer software is made accessible to the public with the consent of the proprietor;

(iii) copying by any means or in any form the computer software;

(iv) using the computer program to produce the same or a substantially similar computer program or a program description of the computer program or of a substantially similar computer program;

(v) using the program description to produce the same or a substantially similar program description or to produce a corresponding computer program;

(vi) using the computer program or a computer program produced as described in (iii), (iv) or (v) to control the operation of a machine having information-processing capabilities, or storing it in such a machine;

(vii) offering or stocking for the purpose of sale, hire or license, selling, importing, exporting, leasing or licensing the computer software or computer software produced as described in (iii), (iv) or (v);

(viii) doing any of the acts described in (vii) in respect of objects storing or reproducing the computer software or computer software produced as described in (iii), (iv) or (v).

Section 6

Infringements

(1) Any act referred to in Section 5(i) to (viii) shall, unless authorized by the proprietor, be an infringement of the proprietor's rights.

(2) The independent creation by any person of computer software which is the same as, or substantially similar to, the computer software of another person, or the doing of any act referred to in Section 5(i) to (viii) in respect of such independently created computer software, shall not be an infringement of the rights of the latter under this Law.

(3) Any presence of the computer software on foreign vessels, aircraft, spacecraft or land vehicles, temporarily or accidentally entering the waters, airspace or land of this country, and any use of computer software during such entry, shall not be considered an infringement of the rights under this Law.

Section 7

Duration of Rights

(1) The rights under this Law shall begin at the time when the computer software was created.

(2)(a) Subject to paragraph (b), the rights under this Law shall expire at the end of a period of 20 years calculated from the earlier of the following dates:

(i) the date when the computer program is, for purposes other than study, trial or research, first used in any country in controlling the operation of a machine having information-processing capabilities, by or with the consent of the proprietor;

(ii) the date when the computer software is first sold, leased or licensed in any country or offered for those purposes.

(b) The rights under this Law shall in no case extend beyond 25 years from the time when the computer software was created.

Section 8

Relief

(1) Where any of the proprietor's rights have been, or are likely to be, infringed, he shall be entitled to an injunction, unless the grant of an injunction would be unreasonable having regard to the circumstances of the case.

(2) Where any of the proprietor's rights have been infringed, he shall be entitled to damages or such compensation as may be appropriate having regard to the circumstances of the case.

Section 9

Application of Other Laws

This Law shall not preclude, in respect of the protection of computer software, the application of the general principles of law or the application of any other law, such as the Patent Law, the Copyright Law or the Law of Unfair Competition.

Table C-1

Entry and Exit Rates for Selected Low Overhead, Easy Entry Industries

	London Cartage 1966-1977	London Pizza 1966-1976	London Pizza 1973-1976	London Printing 1966-1976	London Printing 1973-1976	Kitchener Cartage 1966-1977
1. Average no. entrants	6.0	6.1	7.3	8.3	8.3	1.0
2. As a % of total	11.8%	18.0%	15.5%	12.8%	11.0%	4.3%
3. Average no. exits	5.4	3.5	5.0	5.5	6.3	1.7
4. As a % of total	11.0%	10.5%	10.8%	8.7%	8.5%	7.7%
5. Average no. entrants exiting after one year	2.0	1.9	3.7	1.9	3.0	0.3
(5) ÷ (1)	33.3%	31.1%	50.7%	22.8%	36.1%	30.0%

(cont'd)

Table C-1 (cont'd)

	Toronto Cartage 1970-1976	Thunder Bay Cartage 1966-1978	Windsor Cartage 1967-1977	Ottawa Cartage 1966-1977	Sarnia Cartage 1966-1977
1. Average no. entrants	74.7	2.8	5.9	17.3	2.4
2. As a % of total	16.3%	12.2%	13.9%	15.2%	16.3%
3. Average no. exits	79.0	2.7	6.5	13.5	2.4
4. As a % of total	17.2%	11.9%	15.3%	11.8%	16.3%
5. Average no. entrants exiting after one year	31.0	1.0	2.3	5.2	0.9
(5) ÷ (1)	41.5%	37.0%	39.0%	30.1%	37.5%

SOURCE: Palmer & Engelhart, 1980, p. 24

Appendix D

AN ALTERNATIVE EXPLANATION OF THE METHOD OF ESTIMATING

Suppose that copyright protecton would reduce the costs of developing and marketing computer software, but it is not known by how much. In terms of Figure 4, the location of S' is unknown and hence the height of the trapezoid is also unknown. Suppose, however, that the distance from Q^* to Q', i.e. ΔQ, is known, along with P, Q and the elasticity of demand, E. In a reasonably competitive market, the price will be competed downward by an amount equal to the shift of the supply curve from S to S'. In terms of Figure D.1, P^*, Q^*, Q' and the elasticity of the demand curve are known. Using the assumption of competition, it is clear that the new equilibrium between supply and demand lies above Q' on the demand curve at point F. Consequently the new supply curve, S', must go through that point; by extending a line leftward from F to the vertical axis, P', the new price can be determined and the area of the trapezoid, P^*GFP', can be measured. If, however, the elasticity of demand has been overstated (this overestimation is shown geometrically by drawing a demand curve through G which is flatter than the correct one, D'), the cost reduction (downward shift in the supply curve) and the benefits of copyright will be underestimated. In Figure D.2, this phenomenon is shown by comparing trapezoid P^*GFP' (obtained using too high an estimate of E) with P^* G F' P''. Similar geometry can be used to show the effect of underestimating ΔQ. More simply, direct examination of equation (6) shows that B will be underestimated if the value ΔQ is too low, if P^* is too low, if E is too high and/or if Q^* is too high. The same results emerge from partial differentiation of B with respect to each of the variables in equation (6).

Figure D.1

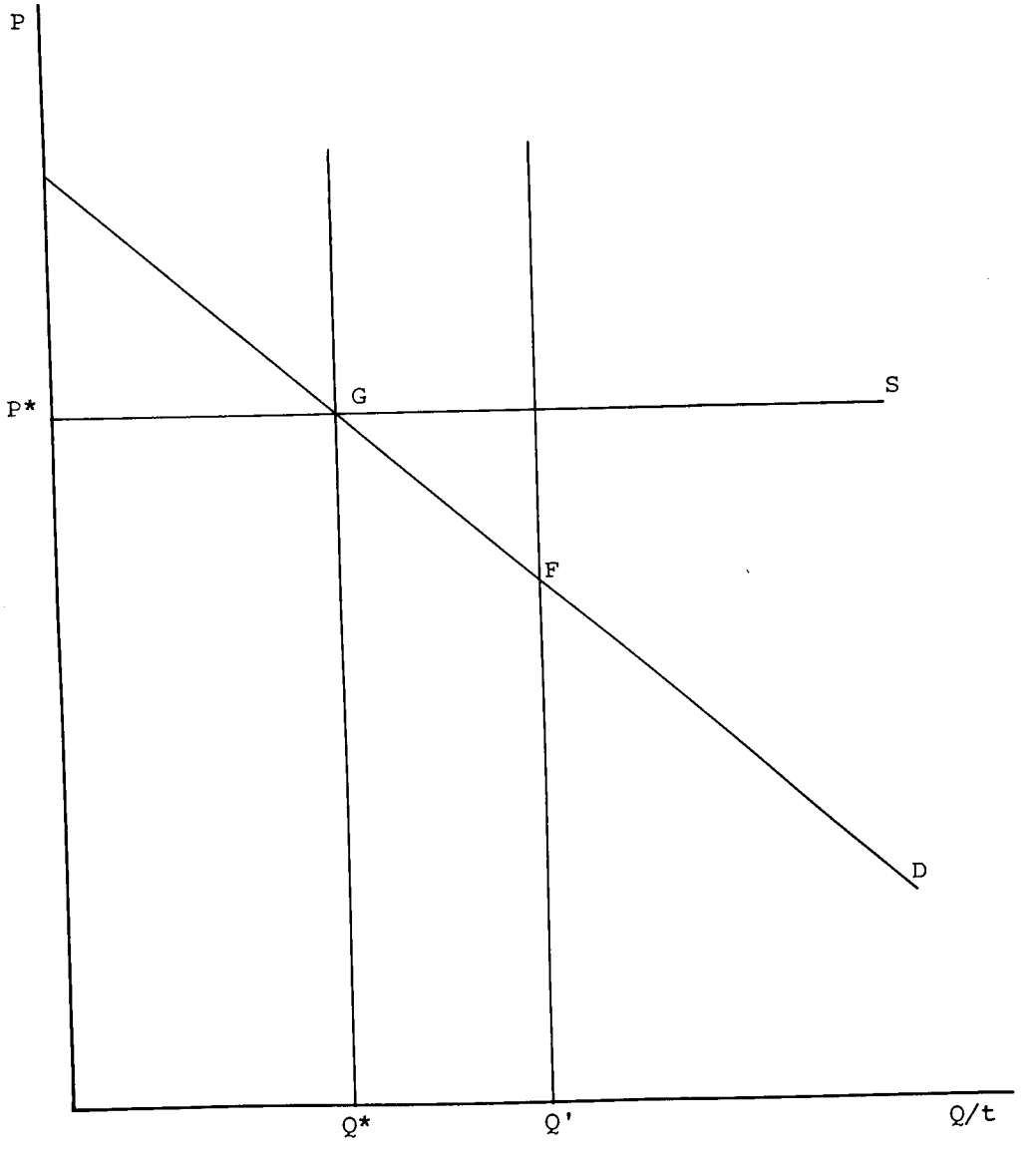
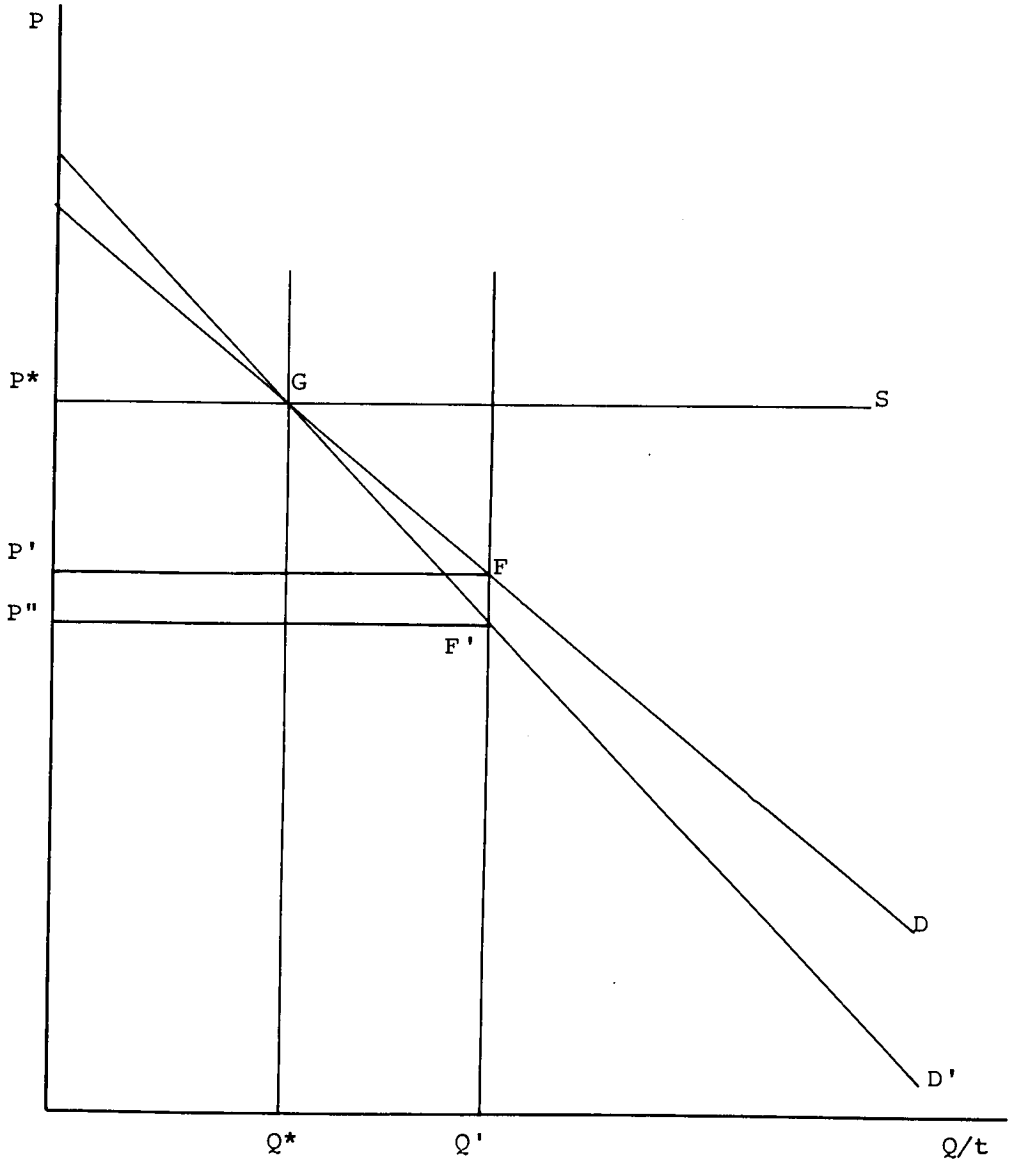


Figure D.2



BIBLIOGRAPHY

Anderson, M. "Algorithm." Encyclopedia of Computer Science and Technology. ed., Belzer, J.; Holzman, A.G.; and Kent, A. New York: M. Dekker, 1975.

Association of Data Processing Service Organizations (ADAPSO). Industry Report. New York, 1978(a).

_____. Brief filed in Parker v. Flook. (437 U.S. 584 [1978]) 1978(b).

Banzhaff, D. "Copyright Protection for Computer Programs." Columbia Law Review, 1964.

Barrigar, R. "Legal Protection of Software from Unauthorized Use." Computers and the Law -- Emerging Issues. Canadian Bar Association, 1976.

Batten, Joseph. Joe Batten's Book: The Story of Sound Recording. London: Rockliff Pub. Corp., 1956.

Benjamin, A. and Kormendi, R. "The Interrelationship Between Markets for New and Used Durable Goods." Journal of Law and Economics, October 1974.

Blomqvist, A.G. and Lim, C. Copyright, Competition and Canadian Culture: The Impact of Alternative Copyright Act Import Provisions on the Book Publishing and Sound Recording Industries. Ottawa: Consumer and Corporate Affairs Canada, 1981.

Braunstein, Y.; Fisher, D.; Ordover, J.; and Baumol, W. Economics and Property Rights as Applied to Computer Software and Data Bases. Springfield, Va.: U.S. Dept. of Commerce, 1977.

Breyer, C. "The Uneasy Case for Copyright: A Study of Copyright in Books, Photocopies and Computer Programs." 84 Harvard Law Review, 1970.

Brock, G.W. The U.S. Computer Industry: A Study of Market Power. Cambridge, Mass.: Ballinger, 1975.

Brozen, Yale. "Entry Barriers: Advertising and Product Differentiation." Industrial Concentration: The New Learning. ed., Goldschmid, H.; Mann, M.; and Weston, F. Boston/Toronto: Little, Brown and Co., 1974.

Canada, Copyright Act, R.S.C. 1970, c-30 (or c-55, as amended).

Canada, Industrial Design Act, R.S.C.

Canada, Patent Act, R.S.C.

Canadian Datasystems. December 1972-1975 and January 1977-1980.

Cary, G. "Copyright Registration and Computer Programs." Copyright Society Bulletin, August 1964.

Cheung, Steven. Property Rights in Trade Secrets. University of Washington, discussion paper, July 1979.

Computer Programs. Cir. 31D. Washington, D.C.: Copyright Office, Library of Congress, 1965.

Computer Software Subcommittee on New Technological Uses of Copyrighted Works (CONTU). Software Subcommittee Report and Additional Views. Washington, D.C.: U.S. Dept. of Commerce, 1977.

_____. Final Report. Washington, D.C.: U.S. Dept. of Commerce, 1978.

Davis, A. "Computer Programs and Subject Matter Patentability." 6 Rutgers Journal of Computers, Technology and the Law, 1979.

Demsetz, Harold. "Toward a Theory of Property Rights." American Economic Review, May 1967.

_____. "Information and Efficiency." Journal of Law and Economics, April 1969.

_____. "Two Systems of Belief About Monopoly." Industrial Concentration: The New Learning. ed., Goldschmid, H.; Mann, M.; and Weston, F. Boston/Toronto: Little, Brown and Co., 1974.

Douglas, E. "Some Ideas on Computers and the Law." Texas Southern University Law Review, 1973.

Economic Council of Canada. Report on Intellectual and Industrial Property. Ottawa, 1971.

Federal Computer Systems Protection Act of 1979, S. 240, 96th Congress, 1st Sess. 83(a) (1979), 125 Cong. Rec. 5711 (Jan. 25, 1979) proposed S 1028(c)(8) to be added to 18 U.S.C. chap. 47.

Fogo, J.G. "Trade Secrets." Proceedings of the Programme of Industrial and Intellectual Property. Law Society of Upper Canada, 1971.

Fox, Harold G. The Canadian Law of Copyright and Industrial Designs. 2nd ed. Toronto: Carswell, 1967.

_____. The Canadian Law and Practice Relating to Letters Patent for Inventions. 4th ed. Toronto: Carswell, 1969.

Frank, W. "1978 DP Budgets." Datamation, January 1978.

_____. "The New Software Economics" (Parts I, II, III and IV). Computerworld, 1979 (January 8, 15, 22 and 29).

Gelatt, Roland. The Fabulous Phonograph: From Edison to Stereo. rev. ed. London: Cassel, 1966.

Gemignani, M.C. "Legal Protection for Computer Software: The View from '79." 7 Rutgers Journal of Computers, Technology and the Law, 1980.

Hindley, B.V. The Economic Theory of Patents, Copyrights and Registered Industrial Designs. Ottawa: Economic Council of Canada, 1971.

Hirshleifer, J. "The Private and Social Value of Information and the Reward to Inventive Activity." 11 American Economics Review, September 1971.

Horowitz, E. and Sahni, S. Fundamentals of Computer Algorithms: Design and Analysis. (Computer Software Engineering Series). Washington, D.C.: Computer Science Press, Inc., 1978.

Hurt, R.M. and Schuchman, R.M. "The Economic Rationale of Copyright." American Economic Review, May 1966.

Kendall, Robert C. Management Perspectives in Programs, Programming and Productivity. Unpublished paper prepared for IBM.

Keon, Jim. A Performing Right for Sound Recordings: An Analysis. Ottawa: Consumer and Corporate Affairs Canada, 1981.

Keyes, A.A. and Brunet, C. Copyright in Canada: Proposals for a Revision of the Law. Ottawa: Consumer and Corporate Affairs Canada, 1977.

Ladas, Stephen P. The International Protection of Industrial Property. Cambridge, Mass.: Harvard Press, 1930.

Liebowitz, S.J. "Copyright in Modern Times." Mimeo. London, Ont.: University of Western Ontario, 1980(a).

_____. "Property Rights and Durable Goods." Mimeo. London, Ont.: University of Western Ontario, 1980(b).

McFadden, E. and Suver, F. "Costs and Benefits of a Data Base System." Harvard Business Review, January-February 1978.

Machlup, Fritz. Economic Review of the Patent System. Washington, D.C.: U.S. Government Printing Office, 1958.

Melville, Leslie. "Computer Software and the Relevance of Copyright." European Intellectual Property Review, November, 1980.

Miller, Richard. Legal Aspects of Technology Utilization. Lexington, Mass.: Lexington Press, 1974.

_____. Legal Protection of Computer Software. New York: Harbridge House Inc., 1977.

_____. "The CONTU Software Protection Survey." 18 Jurimetrics, 1978.

Miller, Roger L. Intermediate Microeconomics. Toronto: McGraw-Hill, 1978.

Morgan, Richard. Computer Contracts. London: Oyez Publishing, 1979.

Nelson, S. "Copyrightability of Computer Programs." 7 Arizona Law Review, 1966.

Niblett, B. and Anderson, M. Memorandum of Evidence submitted to the Whitford Committee. Unpublished, 1975.

Novick, M. and Wallenstein, H. "The Algorithm and Computer Software Patentability: A Scientific View of a Legal Problem." 7 Rutgers Journal of Computers, Technology and the Law, 1980.

Oberman, Michael S. "Copyright Protection for Computer-Produced Directories." 41 Fordham Law Review, May 1974.

Palmer, John and Engelhart, Kenneth. Ontario Cartage Regulation. Ontario Economic Council, unpublished, 1980.

Patent Office Record. Ottawa, August 1, 1978.

Patterson, Lyman Ray. Copyright in Historical Perspective. Nashville, Tennessee: Vanderbilt University Press, 1968.

Plant, Arthur. "The Economic Aspects of Copyright in Books." Economica, 1934.

_____. "The Economic Theory Concerning Patents for Invention." Economica, 1934.

Popper, M. "Technology and Programming -- Is it a Problem in Definitions?" 6 Rutgers Journal of Computers and the Law, 1977.

Posner, Richard A. Economic Analysis of Law. Boston/Toronto: Little, Brown and Co., 1972.

_____. "The Social Costs of Monopoly and Regulation." Journal of Political Economics, August 1975.

Puckett, A.W. "The Limits of Copyright and Patent Protection for Computer Programs." 16 ASCAP, Copyright Law Symposium. New York: Columbia University Press, 1968.

Ralston, A. and Meek, C., eds. Encyclopedia of Computer Science. New York: Petrocelli/Charter, 1976.

Read, Oliver, and Welch, W.L. From Tin Foil to Stereo. Indianapolis: H.W. Sams, 1959.

Report of the President's Commission on the Patent System, 1966.

Ross, O.B. "The Patentability of Computer Firmware." 59 VPOS, 1977.

_____. "The Patentability of Software and Firmware." 3 Computer Law Service S. 4.2 art. 5 (1978).

"A rush of new companies to mass-produce software." Business Week, September 1, 1980.

Saving, T.J. "Estimation of Optimum Size of Plant by the Survivor Technique." Quarterly Journal of Economics, vol. 75, November 1961.

Scherer, F.M. Industrial Market Structure and Economic Performance. 2nd. ed. Chicago: Rand McNally, 1979.

Share Inc. Data Processing in 1980-1985. New York: Wiley, 1976.

Shepherd, W.G. "What Does the Survivor Technique Show About Economies of Scale?" Southern Journal of Economics, July 1967.

Soma, J.T. The Computer Industry. Lexington, Mass.: Lexington Books, 1976.

Statistics Canada. Computer Service Industry. cat. no. 63-222, 1972 to 1978.

Stedman, J. "Copyright Developments in the United States." A.A.U.P. Bulletin 63, October 1976.

Stigler, G.J. "The Economies of Scale." Journal of Law and Economics, vol. 1, 1958.

Stork, B. "Note, Computer Programs and Proposed Revisions of the Patent and Copyright Laws." Harvard Law Review, 81 (1968).

Tapper, C. Computer Law. London: Longman Group Ltd., 1978.

Torno, Barry. International Considerations of Copyright Law Revision in Canada: International Obligations. Ottawa: Consumer and Corporate Affairs Canada, 1978.

_____. Canadian International Copyright Relations Under the Berne and Universal Copyright Conventions. Ottawa: Consumer and Corporate Affairs Canada, forthcoming.

United States, Copyright Act, 1976, 17 U.S.C.

Weiss, L. "The Survival Technique and the Extent of Suboptimal Capacity." Journal of Political Economics, vol. 72, June 1964.

_____. "The Extent of Suboptimal Capacity: A Correction." Journal of Political Economics, vol. 73, June 1965.

Wincor, Richard. Literary Property. New York: Crown Publishers, 1967.

World Intellectual Property Organization (WIPO). Model Provisions on the Protection of Computer Software. Geneva, 1978.

Canada

1M-CORP-82-1